

От автора

Данная книга стала результатом многолетней работы по технической поддержке пользователей системы проектирования печатных плат Protel 99 SE компании Altium. Благодаря уникальной программной архитектуре, эта система практически не имеет ограничений по расширению своей функциональности, чем превосходит все другие популярные программы. Оригинальная система хранения проектных данных с возможностью множественного доступа к одним и тем же файлам по сети позволяет интегрировать усилия нескольких разработчиков в рамках рабочих групп и обеспечить централизованное управление библиотеками и проектами.

На момент написания данной книги уже вышла и начала поставляться пользователям новая версия системы Protel, названная Protel DXP. Эта версия предназначена для работы только под управлением 32-разрядных операционных систем Windows 2000 и XP и предъявляет повышенные требования к аппаратной платформе. Этот факт косвенно сдерживает переход широких масс разработчиков с версии Protel 99 SE на новую, а значит, данная книга найдет своего читателя.

Автор выражает благодарность лично Филу Логхиду (Phil Loughhead), руководителю отдела разработок компании Altium за предоставление всех материалов, необходимых для написания данной книги.

Автор благодарит своих коллег Масленникова Дмитрия, Новенькова Михаила, Фокина Михаила и Константинова Владимира за помощь по переводу сотен листов оригинальной документации.

Отдельное спасибо следует сказать Владимиру Стешенко, опыт которого по написанию книг по тематике автоматизации проектирования электронных устройств помог автору начать (а главное закончить!) работу над данной книгой.

Автор искренне надеется, что приведенные ниже материалы окажутся полезными широкому кругу читателей. Любые замечания по содержанию данной книги можно направлять по адресу potapoff@eltn.ru.

Глава 1

Инсталляция и начало работы

Что такое система Protel 99 SE?

Настоящая книга представляет собой полное описание программы проектирования печатных плат Protel 99 SE. Книга рассчитана как на начинающих пользователей, так и на опытных специалистов, и содержит полную информацию обо всех модулях программы. Помимо прямых инструкций по использованию базовых функций системы, в книге приведены множество полезных советов, направленных на ускорение процесса проектирования и повышение производительности труда разработчика.

С помощью системы Protel 99 SE обычный настольный компьютер превращается в законченное, мощное, полностью интегрированное средство проектирования печатных плат и управления проектной документацией, имеющее все необходимые инструменты для воплощения проекта, начиная с построения базовой концепции и заканчивая подготовкой к производству.

Ядром системы Protel 99 SE является разработанная компанией Altium (ранее Protel) среда проектирования Design Explorer (рис. 1.1), в которую подключаются все стандартные модули, а также пользовательские модули, разработанные с помощью специального пакета Protel Software Developer Kit. Открытая архитектура программы позволяет самостоятельно наращивать ее функциональность не дожидаясь официальных обновлений. Пользовательский интерфейс среды Design Explorer легко настраивается под нужды разработчика, а большинство рутинных операций может быть выполнено с помощью макросов, написанных на внутреннем языке Client Basic. В системе имеются все необходимые средства разработки таких макросов, например, текстовый редактор, система проверки синтаксиса и отладчик. Все макросы выполняются на уровне системы, а значит с их помощью можно управлять любым процессом в системе, независимо от того, какой модуль системы его содержит и выполняет.

Если пользователь работает над тем или иным документом, система Protel 99 SE будет показывать только те командные меню и панели инструментов, которые относятся к данному типу документов. При переключении на другой документ вид и состав панелей инструментов и меню изменится в соответствии с типом нового документа. Разумеется, что эти инструменты можно также легко переконфигурировать. Таким образом, независимо от того, над каким документом сейчас работает разработчик, разводит ли он печатную плату, подсчитывает ли он стоимость комплектующих в

электронной таблице или выполняет анализ переходных процессов, он всегда имеет под рукой полный и удобный набор всех необходимых инструментов.

Следует отметить, что на момент написания этой книги вышла в свет новая версия системы — Protel DXP. Новый продукт содержит ряд новых функций, направленных на обеспечение разработки сложных печатных плат (версия Protel 99 SE позиционировалась для средних проектов). Однако, он намного более требователен к ресурсам и рассчитан на работу только под управлением ОС Windows 2000 и XP, что делает затруднительным быстрый переход на него подавляющего большинства пользователей. Значит, данная книга обязательно найдет своего читателя, а также будет полезна специалистам, изучающим Protel DXP, так как описывает единую идеологию продуктов Protel.

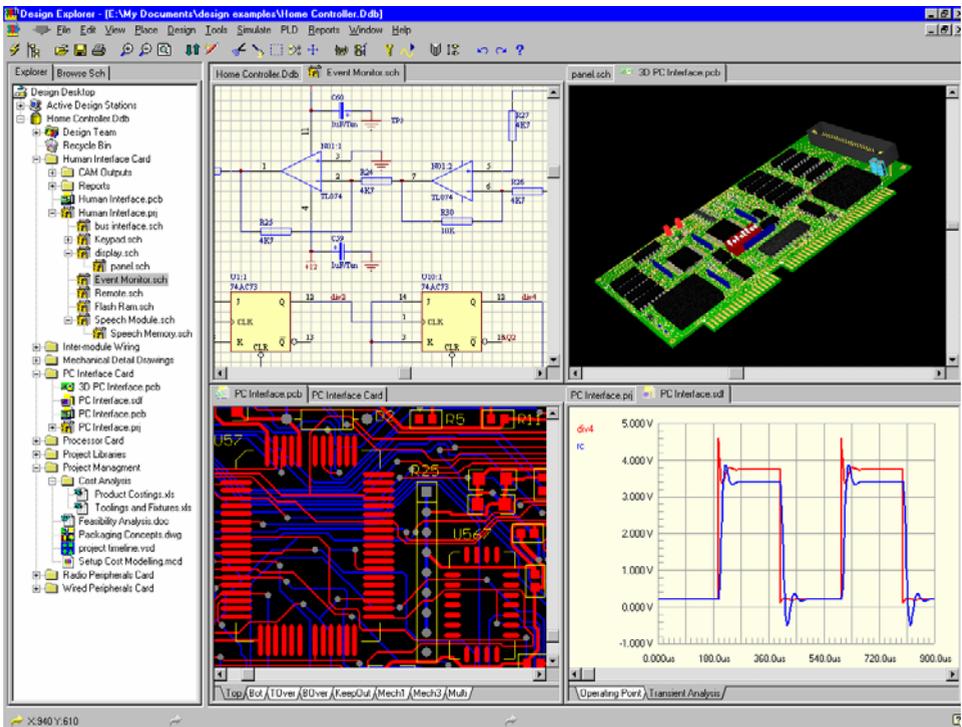


Рис. 1.1 Среда проектирования Design Explorer.

Инсталляция

Системные требования

Минимальные требования

- Операционная система Microsoft Windows 95/98/NT на IBM-совместимом компьютере.
- Процессор Pentium.
- ОЗУ объемом 32 Мбайт.
- SVGA монитор, 16 цветов, разрешение 1024x768 точек.
- 200 Мбайт свободного места на жестком диске (при минимальной инсталляции).

Рекомендуемые требования

- Microsoft Windows NT 4 (или выше)
- Процессор Pentium II 300 MHz.
- ОЗУ объемом 64 Мбайт.
- True color монитор, разрешение 1024x768 точек и выше.
- 300 Мбайт свободного места на жестком диске (для полной инсталляции со всеми библиотеками).

Что входит в комплект поставки системы Protel 99 SE?

В стандартный комплект поставки системы Protel 99 SE входят::

- Компакт-диск и дистрибутивом программы.
- Справочник разработчика (Designer's Handbook).
- Письмо с лицензионным кодом (может доставляться отдельно).
- Регистрационная карточка пользователя (если продукт не поставляется как обновление).
- Лицензионное соглашение.(отрывное приложение к регистрационной карточке).

Если что-либо из выше перечисленного в комплекте отсутствует, немедленно свяжитесь с региональным представительством компании Altium для выяснения причин этого и последующего получения недостающих материалов.

Инсталляция системы Protel 99 SE

В общем случае, программа инсталляции запустится автоматически, сразу после того, как вы вставите диск с дистрибутивом программы в дисковод вашего компьютера.

Если программа инсталляции не запустилась автоматически, запустить ее можно вручную, для чего в меню Windows **Пуск** » **Выполнить** надо набрать командную строку:

D:\Protel99SE\Setup.exe

где D — обозначение устройства чтения CD-ROM вашего компьютера. Далее рекомендуется следовать всем инструкциям, приводимым в последовательно появляющихся диалоговых окнах. В одном из таких окон от пользователя потребуются ввести лицензионный код, при работе в локальной сети многопользовательский лицензионный код может быть получен автоматически с любого из соседних компьютеров с запущенной компанией Protel 99 SE, для чего следует включить опцию **Floating License**.

Последнее диалоговое окно сообщит вам, что для окончательного завершения процесса инсталляции и начала работы с системой Protel 99 SE, необходима перезагрузка вашего компьютера. В противном случае программа работать не будет.

Если при попытке редактировать тот или иной документ система показывает диалоговое окно **Lock**, это означает, что лицензионный код, деблокирующий данный модуль программы, введен неправильно. Проверить существующие права доступа к различным модулям программы можно в диалоговом окне **Security Locks**, выполнив команду меню **Client Menu** » **Security**. Для этого надо с помощью мыши выбрать нужный модуль из списка и нажать кнопку **Unlock**, после чего система покажет вам введенный лицензионный код и соответствующие ему права доступа.

После инсталляции программы рекомендуется заполнить, подписать и отправить по указанному адресу или факсу регистрационную карточку пользователя. Эта карта сообщает компании Altium, что у нее появился очередной лицензионный пользователь, а самому пользователю гарантирует полный доступ к материалам службы технической поддержки компании.

Интерактивная справочная система

Система Protel 99 SE имеет интерактивную справочную систему, оперативно предоставляющую пользователю всю необходимую информацию на интересующую его тему. Для облегчения поиска информации в справочной системе разработан так называемый Помощник (**Help Advisor**), вызываемый командой меню **Help » Search for Help on** (рис. 1.2).

Для того, что бы найти интересующую вас информацию достаточно написать свой вопрос в специальном текстовом поле аналогично тому, как вы спрашивали бы живого человека. После нажатия кнопки **Search** система высветит список разделов справочной системы, имеющих отношение к заданному вами вопросу. Разумеется, справочная система поддерживает только английский язык, поэтому вопрос должен быть задан только на английском языке.

- Для получения оперативной подсказки относительно текущей выполняемой операции и активного объекта нажмите клавишу **F1**.

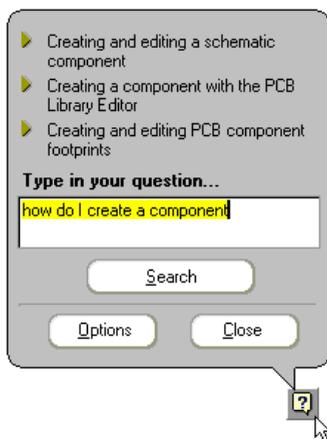


Рис. 1.2. Для вызова Помощника (Help Advisor) необходимо щелкнуть на кнопке со знаком вопроса, расположенной в строке состояния.

Глава 2

Работа в среде проектирования Design Explorer

Среда проектирования Design Explorer

Когда вы запускаете систему Protel 99 SE через меню Windows Пуск » Программы, вы запускаете главный модуль программы Design Explorer, представляющий собой среду проектирования (рис. 2.1). Этот модуль является главным модулем системы, обеспечивающим интерфейс со всеми остальными ее частями и инструментами.

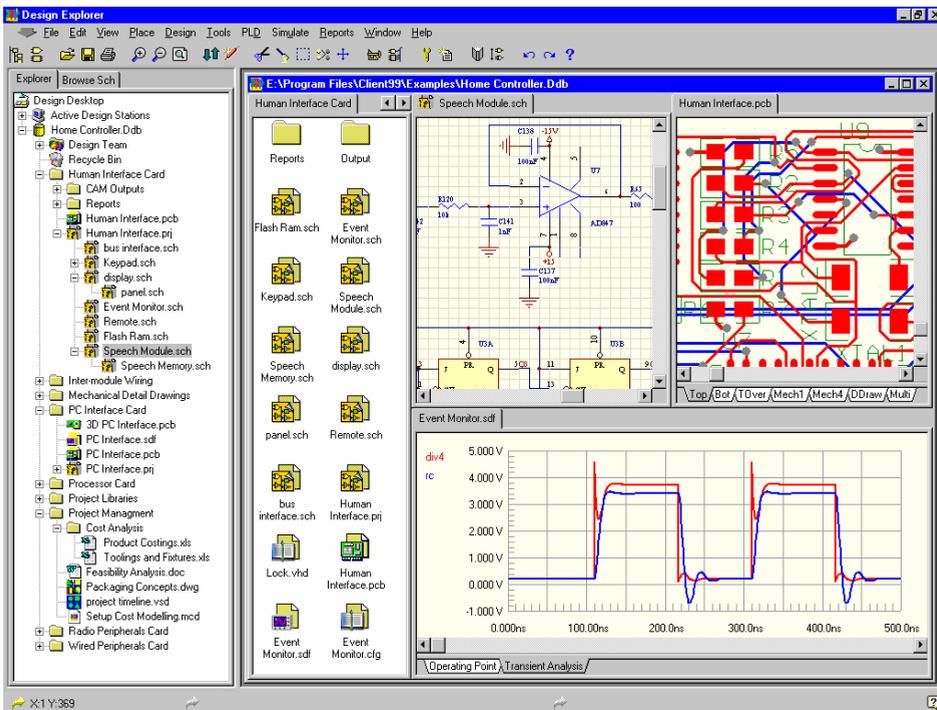


Рис. 2.1. Среда Design Explorer — это интерфейс между пользователем и базой данных проекта

Среда Design Explorer имеет ряд особенностей, отличающих эту программу от других приложений системы Windows:

Здесь имеется возможность сохранять различные документы (файлы) в единой базе данных проекта (Design Database). Это могут быть как стандартные документы системы Protel, например, листы принципиальных схем, чертежи печатных плат, так и файлы созданные другими приложениями системы Microsoft Windows.

Каждый проект в среде Design Explorer открывается как отдельное окно, называемое окном проекта (Design Window). При обращении к тому или иному документу, сохраненному в базе данных проекта, он открывается на отдельной вкладке в рамках соответствующего окна проекта. Это позволяет пользователю легко управлять различными частями нескольких одновременно открытых проектов разной сложности, начиная от простых принципиальных схем и заканчивая сложными многострочными иерархическими проектами с чертежами печатных плат, а также другими документами, например, текстовыми файлами Microsoft Word и электронными таблицами Microsoft Excel, проектами Visio.

Для быстрого управления проектами предназначена панель просмотра дерева проекта (Navigation Panel), расположенная в левой части среды проектирования. С помощью этой панели вы можете без труда перемещаться между проектами, строить иерархические структуры любой глубины и сложности, производить все стандартные операции с документами: копирование, вставку, перемещение и удаление.

Как работать в среде Design Explorer?

Работать в среде Design Explorer чрезвычайно просто. Эта оболочка практически ни-

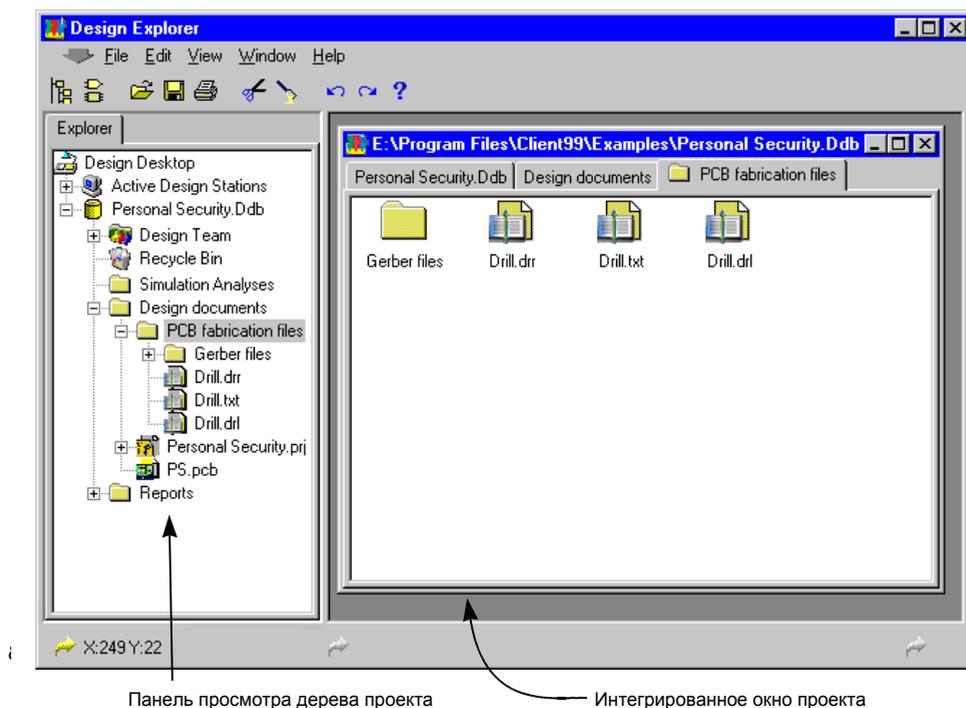


Рис. 2.2. Внешне среда Design Explorer очень похожа на Проводник системы Windows.

чем не отличается от всем известного Проводника системы Windows, а значит, пользователя почти не надо обучать.

Аналогично Проводнику системы Windows здесь имеется панель просмотра дерева проекта слева и поле просмотра активной части проекта справа (рис. 2.1). В среде Design Explorer в этом поле открываются окна Design Window, соответствующие открытым в настоящий момент проектам.

Использование панели просмотра дерева проекта

Панель просмотра дерева проекта (Navigation Panel) расположена в левой части основного окна среды проектирования. Эта панель в виде ветвей дерева показывает отношения между различными частями проекта. Аналогично Проводнику системы Windows здесь показывается, как и где в базе данных проекта сохраняются документы различных типов.

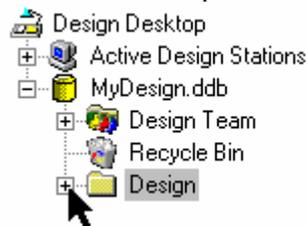
Также панель просмотра ясно показывает всевозможные логические связи между различными листами многолистовых и иерархических проектов.

Работая с панелью просмотра дерева проекта пользователь может (рис. 2.3):

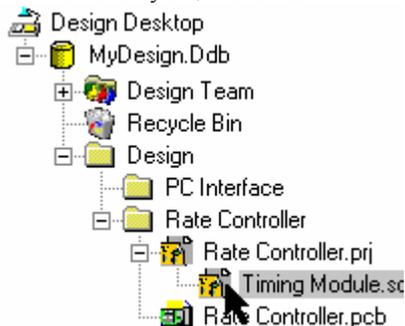
- Открыть очередную ветвь дерева проекта, щелкнув левой кнопкой мыши на значке +.
- Скрыть развернутую ветвь дерева проекта, щелкнув левой кнопкой мыши на соответствующем значке -.
- Выполнив двойной щелчок левой кнопкой мыши на папке на дереве проекта, развернуть ее. Одновременно с этим папка откроется в соответствующем окне проекта в поле справа.
- Открыть нужный документ, выполнив на нем один щелчок левой кнопкой мыши.
- Щелкнув на документе и удерживая левую кнопку мыши, пользователь может переместить выбранный документ в другую папку. Если эта операция производится с папкой, то перемещаются также и все вложенные документы с сохранением внутренней структуры.
- Щелкнув на документе и удерживая правую кнопку мыши, пользователь может указать на другую папку. После отпускания кнопки мыши появится контекстное меню, предлагающее на выбор различные действия с выделенными документами: копирование, перемещение и создание ярлыка.



Чтобы открыть ветвь дерева проекта, щелкните один раз на значке +



Для перемещения вглубь иерархии проекта продолжайте щелкать по соответствующим значкам +



Чтобы открыть документ щелкните на нем один раз

Рис. 2.3. Основные приемы работы с деревом проекта.

- ◆ Лучший способ работать с базой данных проекта — работать с панелью просмотра дерева проекта.
- ◆ Нажимая на маленькие значки "+" и "-" пользователь может без труда просматривать различные ветви дерева проекта, не открывая при этом лишних окон с папками и документами. Когда нужный документ найден, достаточно просто щелкнуть на нем левой кнопкой мыши, и он откроется на новой вкладке в соответствующем окне проекта.

Работа с окном проекта

Окно проекта предлагает пользователю два принципиально отличающихся способа просмотра. Первым из них является просмотр папок, когда в окне в виде иконок отображаются все документы и вложенные папки, содержащиеся в открытой папке (рис. 2.4). Другим является просмотр документа, когда выбранный документ открывается для редактирования соответствующим редактором системы (рис. 2.5).

Пользователь может использовать оба этих способа как по отдельности, так и одновременно в любой комбинации. Имена всех документов и папок, открытых в окне проекта отображаются на соответствующих вкладках. Чтобы сделать какой-либо из открытых документов активным, достаточно просто щелкнуть левой кнопкой мыши на соответствующей вкладке.

- Просмотр папок в окне проекта полностью аналогичен просмотру папок в Проводнике Windows. Здесь пользователь может:
- Открыть папку или документ, выполнив двойной щелчок левой кнопкой мыши на соответствующей иконке.
- Использовать для выделения папок и документов стандартные для системы Windows способы: выделение нескольких файлов мышью при нажатии клавиш **SHIFT** и **CTRL**, а также выделение файлов в окне охвата.

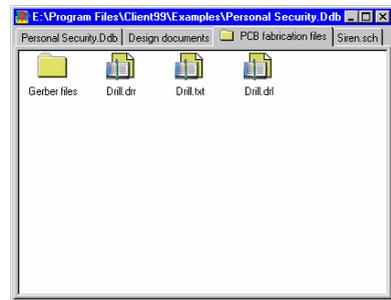


Рис. 2.4. Просмотр содержимого папки в окне проекта.

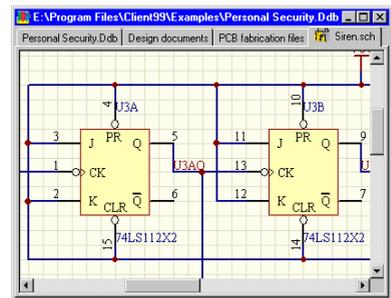


Рис. 2.5. Редактирование документа в окне проекта.

- Удерживая левую кнопку мыши переместить выбранные документы или папки в другую папку, которая может быть открыта даже в другом окне проекта (то есть в другую базу данных проекта).
- Щелкнув на выделенном документе или папке и удерживая правую кнопку мыши, пользователь может указать на другую папку, возможно, в другом проекте. После отпускания кнопки мыши появится контекстное меню, предлагающее на выбор различные действия с выделенными объектами: копирование, перемещение и создание ярлыка.
- Щелчком правой кнопки мыши на иконке папки или документа вызвать контекстное меню, предлагающее операции экспорта и импорта документов форматов, отличных от форматов системы Protel, а также копирования или вырезания выделенного объекта в буфер обмена в рамках системы.
- Щелчком правой кнопки мыши в пустом месте папки вызвать контекстное меню, предлагающее создать новый документ, импортировать документ, вставить связь с внешним документом, вставить объект из буфера обмена или изменить режим просмотра текущего окна.

Для просмотра различных документов служат соответствующие редакторы. Так, например, для просмотра и редактирования листов принципиальных схем вызывается редактор принципиальных схем Schematic Editor, чертежей печатных плат — редактор печатных плат PCB Editor. Текстовые отчеты в формате ASCII редактируются встроенным текстовым редактором Text Editor и так далее. Для того, чтобы перейти из одного редактора к другому, достаточно щелкнуть левой кнопкой мыши по вкладке соответствующего документа в окне проекта, выполнить определенную команду меню или нажать нужную кнопку на панели инструментов.

Работа с несколькими открытыми документами

Главное удобство использования окна проекта заключается в том, что все документы данного проекта будут только в этом окне, а это позволяет избежать путаницы при работе с большими проектами. Последовательный перебор документов окна проекта осуществляется нажатием клавиш **CTRL+TAB** или **CTRL+SHIFT+TAB**, а непосредственный выбор нужного документа — щелчком левой кнопкой мыши на соответствующей вкладке.

Закрытие документов

Чтобы закрыть активный документ необходимо выполнить щелчок правой кнопкой мыши на его вкладке и в появившемся контекстном меню выбрать команду **Close** или выполнить команду меню **File » Close**.

Изменение конфигурации окна проекта

Если пользователю необходимо одновременно просматривать несколько документов в одном окне проекта, то он может разбить окно на несколько секторов, в каждом из которых будет отображаться нужный документ.

Проще всего это сделать выполнив щелчок правой кнопкой мыши на вкладке активного документа (рис. 2.6). Вкладка активного документа отличается от остальных тем, что на ней присутствует маленькая иконка, показывающая тип документа.

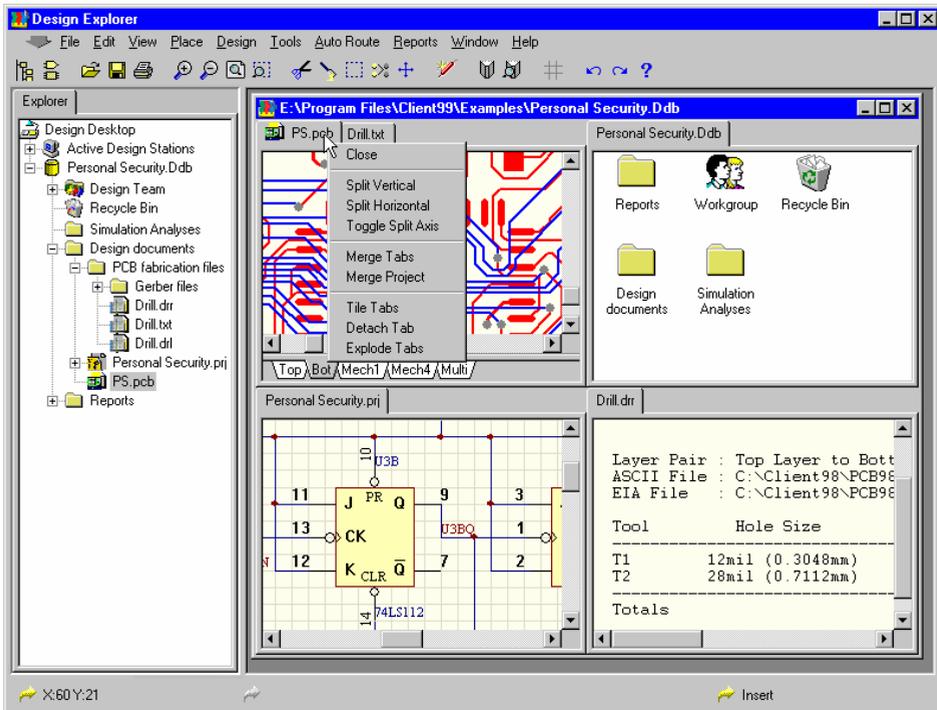


Рис. 2.6. Щелчок правой кнопкой мыши на вкладке с именем документа вызывает всплывающее меню конфигурации окна проекта

Появится контекстное меню предлагающее широкий набор команд управления просмотром документов. На рисунке выше изображен результат действия команды **Tile Tab** (упорядочить мозаикой) над четырьмя вкладками, выбранными на момент вызова меню.

Изменение положение того или иного подокна в окне проекта

Изменить местоположение того или иного подокна можно выполнив щелчок левой кнопкой мыши на соответствующей вкладке и удерживая ее переместить подокно в нужный сектор окна проекта. После нескольких подобных упражнений пользователь без труда сможет управлять расположением документов.

Изменение размера секторов в окне проекта

Если направить указатель мыши на границу, разделяющие сектора в пределах одного окна проекта, то указатель мыши изменит свой вид на две маленькие стрелки, направленные в противоположные стороны. Далее надо щелкнуть на границе левой кнопкой мыши и, удерживая ее, передвинуть границу в нужном направлении.

Отказ от разбиения окна проекта на сектора

Чтобы отказаться от разбиения окна на сектора и вернуться в режим просмотра только одного активного документа, необходимо выполнить щелчок правой кнопкой мыши на любой из вкладок документов и в появившемся контекстном меню выбрать команду **Merge Tabs**.

Отображение документа в отдельном окне

Для того, чтобы отобразить какой-либо документ в отдельном окне вне границ окна проекта, требуется выполнить щелчок правой кнопкой мыши на соответствующей вкладке и в появившемся контекстном меню выбрать команду **Detach Tab**.

Возвращение документа, отображаемого в отдельном окне, в окно проекта

Для того, чтобы вернуть отображаемый в отдельном окне документ в рамки окна проекта, нужно выполнить щелчок правой кнопкой мыши на его вкладке и в появившемся контекстном меню выбрать команду **Merge Project**.

Что такое база данных проекта?

В тексте данной главы термин "база данных проекта" (Design Database) уже упоминался, и сейчас самое время выяснить, что же он означает.

В различных словарях и справочниках термин "база данных" определяет упорядоченный набор данных с удобным доступом к ним. Это определение как нельзя лучше характеризует то, что в системе Protel 99 SE называется Design Database — набор всех документов проекта, организованных и сохраненных в виде отдельных файлов на жестком диске, с удобным доступом к каждому из них. Для обозначения базы данных проекта в системе Protel 99 SE используется расширение .DDB.

Все документы, созданные в процессе работы с проектом, будь то листы принципиальных схем, чертежи печатных плат, списки используемых материалов, таблицы сверления, выходные файлы в формате Gerber, сохраняются в базе данных проекта.

Внутри нее сохраняются также вложенные папки и информация о иерархических связях.

Когда пользователь впервые открывает базу данных проекта, в окне проекта открывается только папка, соответствующая самому верхнему уровню, но ни один документ внутри нее не будет открыт. Это означает, что даже самый сложный и объемный проект, содержащий тысячи документов, будет открываться очень быстро.

Работая с базой данных проекта, пользователь имеет возможность выборочно открывать те или иные содержащиеся в ней документы. Для того, чтобы открыть документ, достаточно выполнить один щелчок левой кнопкой мыши на иконке документа, отображаемой на дереве проекта, или двойной щелчок на соответствующей иконке в окне проекта. Выбранный документ будет открыт в окне проекта с использованием соответствующего редактора.

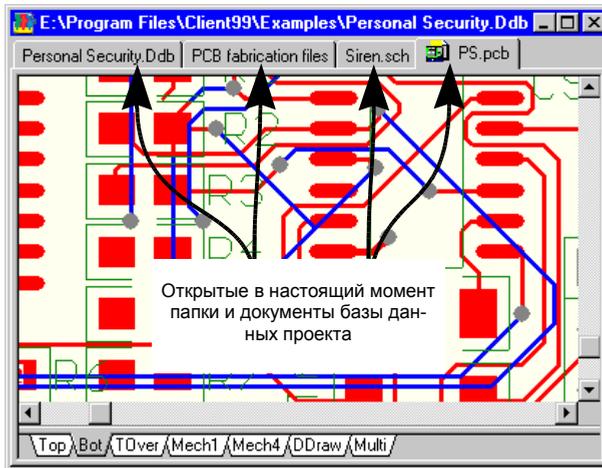


Рис. 2.7. Закладки в нижней части окна проекта показывают, какие документы открыты в настоящий момент

Проверить, какие документы в настоящий момент открыты для просмотра и редактирования, пользователь может перебрав с помощью мыши все представленные в окне проекта вкладки (рис. 2.7).

Создание новой базы данных проекта

Для того, чтобы создать новую базу данных проекта, необходимо выполнить команду меню **File » New Design**. Откроется диалоговое окно создания новой базы данных проекта New Design Database, в котором следует ввести имя проекта и его местоположение на диске (рис. 2.8).

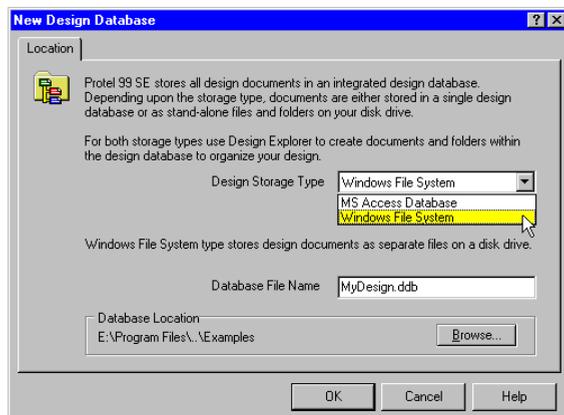


Рис. 2.8. Создание новой базы данных проекта — выберите способ хранения, введите имя и задайте местоположение.

Выбор способа хранения документов

Система Protel 99 SE допускает два различных способа хранения документов внутри проекта:

- При выборе опции MS Access Database все документы проекта будут храниться внутри одного файла базы данных формата Microsoft Access.
- При выборе опции Windows File System все документы хранятся непосредственно на диске в виде файлов и папок.

Не зависимо от выбранного способа хранения документов основные принципы работы с документами проекта в среде Design Explorer остаются одинаковыми. Если проект хранится в виде файловой системы, пользователю все равно придется сначала открыть проект и только потом схему, чертеж печатной платы или текстовый документ. Просто добавлять документы в папки проекта средствами Проводника Windows в этом случае нельзя, так как документы должны быть импортированы в заранее созданную базу данных проекта. Тем не менее, здесь допускается импорт

документов методом перетаскивания (drag-and-drop) непосредственно из Проводника системы Windows.

Заметим, что при хранении документов проекта в виде папок и файлов непосредственно на диске все функции групповой соразработки проекты недоступны, например, функция контроля доступа к документу. Все остальные функции, например, синхронизация и фоновое открытие документов, работают без изменений.

Защита базы данных проекта паролем

Если пользователь желает защитить паролем создаваемую базу данных проекта от несанкционированного доступа, то ему необходимо щелкнуть левой кнопкой мыши по вкладке **Password** и там ввести нужный пароль. Защитить паролем проект пользователь может в любой момент работы на нем, для чего надо войти в папку **Members** и ввести пароль для пользователя **Admin**. Для отмены защиты паролем, наоборот, достаточно удалить пароль для пользователя **Admin**.



Рис. 2.10. Новая база данных проекта, готовая к созданию новых папок и документов

Создание нового документа внутри базы данных проекта

Чтобы добавить новый документ в базу данных проекта, надо в окне с открытой папкой, в которой в дальнейшем будет храниться создаваемый документ, выполнить щелчок правой кнопкой мыши и в появившемся контекстном меню выбрать команду **New**.

Откроется диалоговое окно создания нового документа **New Document**, в котором представлен обширный набор иконок, соответствующий всем доступным типам документов (рис. 2.11). Далее следует выбрать нужный тип документа и нажать кнопку **ОК**.

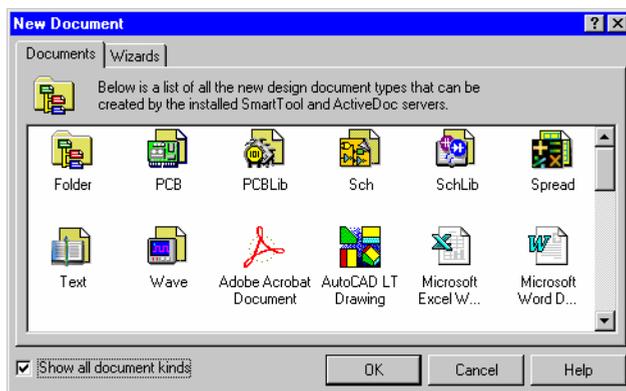


Рис. 2.11. Окно создания нового документа.

Количество доступных типов документов определяется наличием в системе Protel или на компьютере соответствующего редактора. Чтобы просмотреть полный список документов, создание которых возможно внутри базы данных проекта системы Protel, необходимо включить флаг **Show all documents kinds**.

Иконка, соответствующая только что созданному документу, появится в окне с выбранной папкой. Для того, чтобы переименовать документ, необходимо щелкнуть один раз левой кнопкой мыши на его иконке (в окне проекта, а не на панели просмотра дерева проекта) и выполнить команду меню **Edit » Rename** или нажать клавишу F2. Имя документа подсветится и пользователь сможет ввести новое.

Почему в диалоговом окне New Document присутствуют иконки не только стандартных документов системы Protel 99 SE?

Одной из самых мощных функций системы Protel 99 SE является возможность работы не только со стандартными документами системы, но и с любыми другими документами, тип которых зарегистрирован на данном компьютере.

При включении флага **Show all documents kinds** в окне **New Document** появятся иконки, соответствующие всем документам, создание и редактирование которых возможно с использованием OLE сервера.

Документы нестандартных для системы Protel 99 SE типов будут создаваться и храниться внутри базы данных проекта, но редактироваться они будут с помощью приложения (OLE сервера), их создавшего.

Благодаря этой функции, все документы, имеющие даже самое отдаленное отношение к данному проекту, сохраняются в единой базе данных проекта. Выгода здесь очевидна: проект легко переносить, все документы хранятся вместе, доступ к любым из них при групповой разработке осуществляется и ограничивается только средствами системы Protel. Механизм разграничения прав доступа подробно описан в разделе *Настройка прав доступа при групповой разработке проекта* данной главы.

Редактирование документов

Когда пользователь впервые открывает базу данных проекта, в окне проекта открывается только папка, соответствующая самому верхнему уровню, но ни один документ внутри нее не будет открыт. Имеется два способа открыть документ для редактирования.

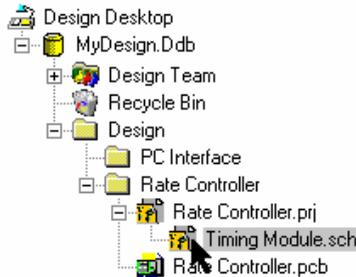


Рис. 2.12. С помощью панели просмотра очень легко находить и открывать документы

Первым способом документ открывается посредством выполнения одного щелчка левой кнопкой мыши на соответствующей иконке на панели просмотра дерева проекта. Преимущество этого способа заключается в том, что пользователь может быстро открыть нужный документ не зависимо от того, как глубоко он сохранен в проекте, то есть не открывая все промежуточные папки. Работа с панелью просмотра дерева проекта подробно описана в разделе *Использование панели просмотра дерева проекта* данной главы.

Второй способ заключается в выполнении двойного щелчка левой кнопкой мыши на иконке, соответствующей нужному документу, в окне проекта. Однако, если нужный документ хранится глубоко в проекте, то потребуются последовательно открыть все промежуточные подпапки, что не всегда удобно.

Редактирование документов, не являющихся стандартными для системы Protel 99 SE

Так как имеется возможность создавать и сохранять в базе данных проекта документы типов, отличных от стандартных для системы Protel, то и имеется возможность их редактировать.

Документы, созданные с помощью OLE сервера, например, документы Microsoft Word и Excel, редактируются непосредственно из среды проектирования Design Explorer. Если пользователь щелкнет на документе Word, созданном и сохраненном внутри базы данных проекта, то запустится приложение Microsoft Word где этот документ можно отредактировать. Все сделанные изменения в последствии будут сохранены в базе данных проекта.

Импорт внешних документов в базу данных проекта

Чтобы импортировать документ в базу данных проекта, надо в окне с открытой папкой, в которой в дальнейшем будет храниться импортируемый документ, выполнить щелчок правой кнопкой мыши и в появившемся контекстном меню выбрать команду **Import**. Появится диалоговое окно **Import Document**, в котором необходимо выбрать нужный документ и нажать кнопку **Open**, после чего копия документа будет сохранена внутри базы данных проекта, а в соответствующую папку будет добавлена иконка.

Иконка показывает, какой редактор будет использоваться при открывании и редактировании только что импортированного документа. Например, документ с чертежом печатной платы в формате Protel PCB будет показан с помощью соответствующей иконки, а значит и открываться для редактирования будет редактором печатных плат PCB Editor. Документы Microsoft Word будут иметь всем известную иконку и открываться с помощью этого редактора. Чертежи системы AutoCAD с расширением .DWG будут показаны иконкой, принятой в системе AutoCAD. При двойном щелчке левой кнопкой мыши на такой иконке система AutoCAD запустится вне среды проектирования Design Explorer и загрузит выбранный файл.

Если система Protel не распознала, каким редактором будет открываться и редактироваться только что импортированный документ, то этот он будет показан специальной иконкой, подсказывающей пользователю, что данный документ нельзя открыть двойным щелчком левой кнопкой мыши. Чтобы изменить тип редактора, сопоставленному данному типу документов, надо выполнить щелчок правой кнопкой мыши на нужном документе и в появившемся контекстном меню выбрать команду **Properties**.

Импортировать документы и папки в базу данных проекта можно и перетаскиванием из Проводника системы Windows, используя технологию drag-and-drop. Все вложенные папки и файлы будут импортированы в проект с учетом изначальной иерархии и станут доступными в среде Design Explorer. Заметим, что при импорте документа его оригинальная копия остается храниться на диске компьютера. Не следует также увлекаться импортом в проект файлов большого размера, например, графических изображений, так как это сильно увеличивает объем базы данных проекта и снижает быстродействие системы.

Добавление связей с внешними документами

Система Protel позволяет вводить в базу данных проекта связи с внешними документами. Эта функция дает возможность использовать внешние документы без необходимости сохранения их внутри проекта. Связанные документы открываются и редактируются в рамках окна проекта аналогично импортированным документам.

У данного способа работы с документами есть одно явное неудобство. Так как связанные документы хранятся вне базы данных проекта, пользователь должен всегда помнить о них при выполнении каких-либо действий с проектом — резервном копировании, перемещении, копировании, удалении. Управление всеми связанными файлами осуществляется средствами работы с файлами операционной системы Windows. Другим неудобством является невозможность заблокировать доступ другому пользователю к используемому в вашем проекте связанному документу, из-за чего этот документ, а значит, и ваш проект, может быть изменен без вашего ведома.

Чтобы добавить в базу данных проекта связанный документ, нужно в окне с открытой папкой, в которой в дальнейшем будет храниться ярлык связи с внешним документом, выполнить щелчок правой кнопкой мыши и в появившемся контекстном меню выбрать команду **Link**. Появится диалоговое окно **Link Document**, в котором необходимо выбрать нужный документ и нажать кнопку **Open to Link**, после чего в соответствующую папку будет добавлена иконка. Заметим, что данная иконка будет иметь внизу маленькую стрелочку, аналогичную ярлыкам системы Windows, сигнализирующую о том, что это связанный документ (рис. 2.13).

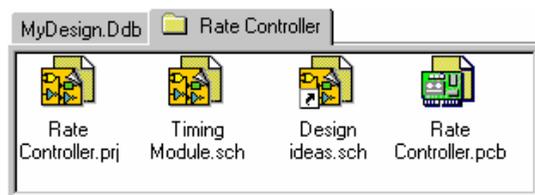


Рис. 2.13. Маленькая стрелка в левом нижнем углу иконки документа Design ideas.sch говорит о том, что он хранится вне базы данных проекта

Экспорт документов из базы данных проекта

В любой момент работы над проектом пользователь имеет возможность выполнить операцию экспорта какого-либо документа из базы данных проекта. Для этого нужно в окне с открытой папкой, в которой находится нужный документ, выполнить щелчок правой кнопкой мыши и в появившемся контекстном меню выбрать команду **Export**. Появится диалоговое окно **Export**, в котором необходимо задать папку на дис-

ке или сетевом устройстве, куда будет сохранен документ, и нажать кнопку **Save**. Заметим, что при выполнении этой операции оригинальная копия документа из базы данных проекта не удаляется, просто в назначенном месте на диске создается его копия.

Экспортировать документы и папки из базы данных проекта можно и перетаскиванием в Проводник системы Windows, используя технологию drag-and-drop. Все вложенные папки и файлы будут скопированы на диск с учетом изначальной иерархии.

Управление документами внутри базы данных проекта

Управление документами, входящими в базу данных проекта в среде проектирования Design Explorer почти ничем не отличается от работы с файлами в Проводнике операционной системы Microsoft Windows.

Использование папок для построения иерархии

Пользователь может систематизировать документы в базе данных проекта посредством создания вложенных папок, группирования и сохранения в них документов. Работа с папками здесь полностью аналогична работе в Проводнике системы Windows (рис. 2.14).

Чтобы создать новую папку в базе данных проекта, надо в окне с открытой папкой, в которой будет создана новая папка, выполнить щелчок правой кнопкой мыши и в появившемся контекстном меню выбрать команду **New**. В появившемся диалоговом окне **New Document** необходимо выбрать иконку **Folder** и нажать кнопку **OK**.

В окне проекта появится новая папка с именем по умолчанию, например, **Folder1**, которое можно изменить, один раз щелкнув левой кнопкой мыши на иконке нужной папки и выполнив команду меню **Edit » Rename** или нажав клавишу **F2**. Имя папки подсветится и пользователь сможет ввести новое.

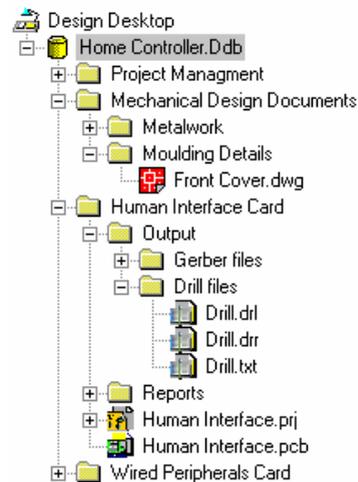


Рис. 2.14. Папки используются для упорядочивания хранения документов внутри базы данных проекта

Перемещение, копирование и удаление документов

Среда проектирования Design Explorer поддерживает все операции с файлами, доступные в Проводнике системы Windows.

- Здесь возможно выделение документов в окне охвата, а также посредством одновременного щелчка левой кнопкой мыши и нажатия клавиш **SHIFT** и **CTRL**.
- Для удаления выделенных документов используется клавиша **DELETE**.
- Для перемещения текущего выделения используется перетаскивание (технология drag-and-drop).
- При перемещении текущего выделения с нажатой правой кнопкой мыши и последующем отпускании ее появляется контекстное меню, предлагающее набор действий: копировать (**Copy**), переместить (**Move**) и создать ярлык (**Create Shortcut**).
- Для быстрого копирования выделенных документов используется комбинация клавиш **CTRL+C**, для быстрой вставки — **CTRL+V**.
- Нажатие правой кнопки мыши на выделенном объекте вызывает контекстное меню, предлагающее набор различных действий с объектами: копирование, вырезание, вставка и так далее.

Переименование документа или папки

Чтобы переименовать документ или папку, входящую в базу данных проекта, необходимо щелкнуть один раз левой кнопкой мыши на его иконке (в окне проекта, а не на панели просмотра дерева проекта) и выполнить команду меню **Edit » Rename** или нажать клавишу **F2**. На месте имени документа или папки появится небольшое окошко с выделенным в нем текстом, в котором надо ввести новое имя с расширением. Система Protel 99 SE не использует расширения файлов для идентификации их типа внутри базы данных проекта, а значит, пользователь может назначать какие угодно расширения или вообще не использовать их.

Настройка прав доступа при групповой разработке проекта

Система Protel 99 SE включает технологию SmartTeam, обеспечивающую возможность одновременного доступа к документам одного и того же проекта различным разработчикам при четком разграничении прав доступа и соблюдении конфиденциальности. Эта технология в общих чертах напоминает стандартную систему управления сетевым доступом, где используется система разрешений для папок и файлов.

По умолчанию при создании новой базы данных проекта система разграничения прав доступа не активируется. Ограничение доступа начинает работать только после того,

как пользователь **Admin** задаст пароль для доступа к проекту. После того, как это будет сделано, при любой попытке открыть проект пользователям будет предложено ввести свое имя и пароль доступа. Открыть базу данных проекта пользователь сможет только в случае если он определен в папке **Member**.

Для этого, сразу после создания новой базы данных проекта, рекомендуется открыть папку **Design Team** (рис. 2.15), в ней открыть папку списка пользователей **Members** и в ней определить новых пользователей. После этого необходимо открыть папку разрешений **Permissions** и задать права доступа для каждого из них.

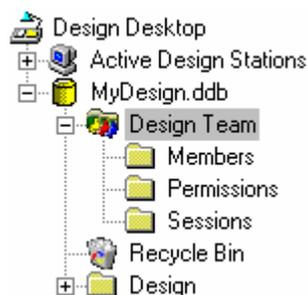


Рис. 2.15. Системная папка Design Team, содержащая инструменты управления доступом к различным частям проекта.

Определение команды разработчиков

При создании новой базы данных проекта в команде разработчиков по умолчанию присутствуют только два пользователя: **Admin** (администратор) и **Guest** (гость). Для того, чтобы создать нового пользователя надо выполнить щелчок правой кнопкой мыши в папке **Members** в окне проекта и в появившемся контекстном меню выбрать команду **New Member** (рис. 2.16).

После этого откроется диалоговое окно **User Properties**, где необходимо указать: **Name** — имя пользователя, которое будет использоваться в окнах **Sessions** и **Permissions**, а также запрашиваться при открытии базы данных; **Description** — описание пользователя и **Password** — индивидуальный пароль доступа.

Обратите внимание, что пользователи и их права доступа могут быть определены только администратором, то есть пользователем с именем **Admin**.

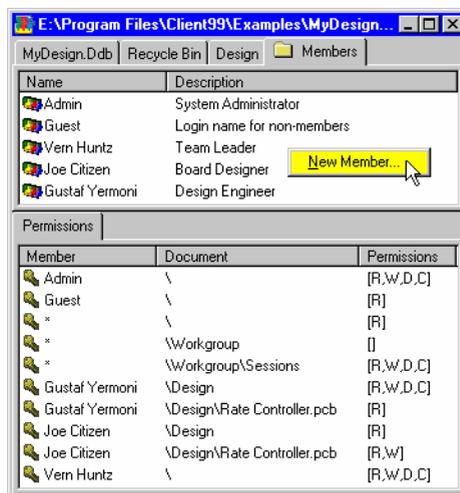


Рис. 2.16. Задание нового члена рабочей группы

Определение разрешений для команды разработчиков

По умолчанию система разграничения прав доступа системы Protel настроена так, чтобы давать пользователям полную свободу для считывания (**Read**), записи (**Write**), создания (**Create**) и удаления (**Delete**) документов, которая будет ограничена введением соответствующих разрешений.

Система разрешений опирается на иерархию проекта — разрешения, действующие на верхнем уровне иерархии могут перекрываться разрешениями на нижнем уровне иерархии.

Обратите внимание, что разрешения могут быть определены только администратором, то есть пользователем с именем Admin.

Разрешения по умолчанию

Разрешения, определяемые по умолчанию при создании нового проекта представлены в таблице 2.1.

Таблица 2.1

Пользователь	Документ	Разрешения	Определение
Admin	\	[R,W,D,C]	Знак "\" означает корневую (самую верхнюю) папку базы данных проекта. Разрешения, приведенные в скобках означают, что пользователю Admin разрешается считывание (R), запись (W), удаление (D) и создание (C) документов во всех папках проекта. Таким образом, пользователь Admin имеет полную свободу действий в любом месте базы данных проекта.
Guest	\	[R]	Разрешения для пользователя Guest означают, что ему разрешается только просмотр, то есть чтение (R), всех документов в проекта.
[All members]	\	[R,W,D,C]	Знак "*" означает, что всем пользователям, за исключением Guest разрешается считывание (R), запись (W), удаление (D) и создание (C) документов во всех папках проекта. Чтобы ограничить доступ к базе данных проекта необходимо оставить разрешение только на чтение (R) или вообще удалить все разрешения.
[All members]	\Design Team	[]	Данная запись означает, что ни один пользователь не имеет доступа к папке с именем Design Team, а также ко всем вложенным в нее папкам. Пользователи смогут видеть папки Members, Permissions и Sessions, но не смогут открыть их. Таким способом можно защитить папки Members и Permissions от изменения любыми пользователями кроме Admin.
[All members]	\Design Team\Sessions	[R,W,D,C]	Данное разрешение означает, что все пользователи имеют полные права доступа к папке Sessions. Данная запись демонстрирует, что разрешения для вложенной папки Sessions перекрывают разрешения для папки Design Team, в которой она создана.

Пример задания разрешений

Приведенный на рисунке 2.17 пример наглядно иллюстрирует процедуру определения прав доступа различным пользователям.

Member	Document	Permissions
Admin	\	[R,W,D,C]
Guest	\	[R]
[All members]	\	[R]
[All members]	\Workgroup	[]
[All members]	\Workgroup\Sessions	[R,W,D,C]
Gustaf Yermoni	\Design	[R,W,D,C]
Gustaf Yermoni	\Design\Rate Controller.pcb	[R]
Joe Citizen	\Design	[R]
Joe Citizen	\Design\Rate Controller.pcb	[R,W]
Vern Huntz	\	[R,W,D,C]

Рис. 2.17. Пример задания разрешений.

Пользователь с именем **Admin** имеет полные права на считывание (**R**), запись (**W**), удаление (**D**) и создание (**C**) документов и папок в любом месте базы данных проекта.

Пользователь с именем **Guest** может только просматривать проект.

Запись **All members** означает, что все пользователи имеют право просматривать любые документы проекта. Установленные по умолчанию права на запись (**W**), удаление (**D**) и создание (**C**) здесь были ограничены. Это также означает, что для того, чтобы расширить права доступа кому-либо, администратору придется описывать дополнительные разрешения.

Пользователь с именем **Gustaf Yermony** имеет полный доступ ко всем папкам и документам, находящимся в папке **Design** и ниже, за исключением документа **Rate Controller.pcb**, который он может только просматривать.

Пользователь с именем **Joe Citizen** может просматривать все документы, расположенные в папке **Design** и ниже, за исключением документа **Rate Controller.pcb**, который он может еще и редактировать.

Пользователь с именем **Vern Huntz** имеет полные права доступа ко всем без исключения папкам проекта.

Мониторинг групповой разработки проекта

В процессе групповой разработки проекта для руководителя важно знать, кто из разработчиков и над каким документом работает в настоящий момент. Для просмотра отчета об открытых в настоящее время документах служит папка **Sessions** (рис. 2.18).

В этой папке отображается список в котором представлены: **Name** — название открытого документа, **Location** — его местоположение в базе данных проекта, **Member** — имя пользователя, **Machine** — сетевое имя компьютера, на котором данный пользователь работает, и **Status** — статус документа.

Приведенный на рисунке 2.19 пример наглядно иллюстрирует процедуру мониторинга рабочей группы.

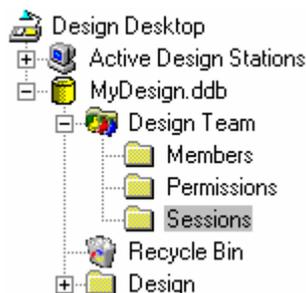


Рис. 2.18. Папка Sessions.

Name	Location	Member	Machine	Status
Rate Controller.prj	\Design\	Gustaf Yermoni	Gustaf_Y	
Timing Module.sch	\Design\	Gustaf Yermoni	Gustaf_Y	Locked
Rate Controller.pcb	\Design\	Joe Citizen	Joe_C	Locked
Rate Controller.pcb	\Design\	Vern Huntz	Vern_H	
Timing Module.sch	\Design\	Vern Huntz	Vern_H	

Рис. 2.19. Мониторинг работы группы разработчиков.

Пользователь с именем **Gustaf Yermoni** в настоящий момент работает с двумя документами: **Rata Controller.prj** и **Timing Module.sch**. Он заблокировал доступ к документу **Timing Module.sch** и собирается заблокировать документ **Rata Controller.prj**. Блокировка означает, что другие пользователи смогут только открыть эти документы, но не смогут сохранить их, даже не смотря на то, что они имеют разрешения на полный доступ к ним.

Пользователь **Joe Citizen** открыл и заблокировал документ **Rate Controller.pcb**.

Пользователь **Vern Huntz** открыл документы **Rate Controller.pcb** и **Timing Module.sch** уже после того, как они были открыты и заблокированы другими пользователями, поэтому он не сможет сохранить сделанные в этих документах изменения не смотря на то, что у него имеются полные права доступа к любым документам проекта.

Для получения самой последней информации о работе в группе необходимо выполнить обновление содержимого папки **Sessions** с помощью команды меню **View » Refresh**.

Блокирование документов

Щелчок правой кнопкой мыши на документе в папке **Sessions** вызывает контекстное меню, предлагающее пользователю заблокировать или разблокировать указанный документ. Если какой-либо документ был однажды заблокирован кем-то из группы разработчиков, то и разблокировать его сможет только он. Когда пользователь закрывает документ, то блокировка автоматически снимается.

Даже заблокированный документ может быть открыт другими пользователями, имеющими соответствующие права доступа. В этом случае на экране их компьютеров появляется сообщение, что данный документ заблокирован таким-то пользователем, и они могут работать с документом только в режиме чтения (**Read-Only**).

Чтобы заблокировать документ, надо в папке **Sessions** выполнить щелчок правой кнопкой мыши на его имени.

Работа с ранее созданными проектами системы Protel

В системе Protel 99 SE все документы, которые пользователь открывает в среде Design Explorer, должны быть сохранены внутри какой-либо базы данных проекта (или быть связаны с ней). Это означает, что пользователь не может просто открыть лист принципиальной схемы или чертеж печатной платы, отредактировать его и затем сохранить на диск. При каждой операции открытия отдельного документа он сначала будет автоматически импортирован в новую базу данных проекта и уже только после этого открыт соответствующим редактором.

Чтобы открыть ранее созданный на диске документ системы Protel 99 SE, необходимо выполнить команду меню **File » Open**, в появившемся диалоговом окне задать тип файлов **All Files (*.*)**, выбрать нужный файл и нажать кнопку **Open**. Как только вы это сделали, система запросит подтверждение на сохранение выбранного файла как базы данных проекта. После задания имени базы данных оригинал документа будет импортирован в нее и открыт для редактирования. Документ, отображаемый при этом в окне проекта будет представлять собой уже не оригинал, а его копию. Для того, чтобы обновить оригинал документа после редактирования потребуется выполнить обратную операцию его экспорта из базы данных проекта.

Преобразование существующего проекта в базу данных проекта

Когда пользователь открывает существующий отдельный документ системы Protel, он сначала будет автоматически импортирован в новую базу данных проекта. Для большего удобства рекомендуется сначала создать новую пустую базу данных проекта, создать в ней необходимую структуру папок и выполнить операцию импорта документа в нужную папку.

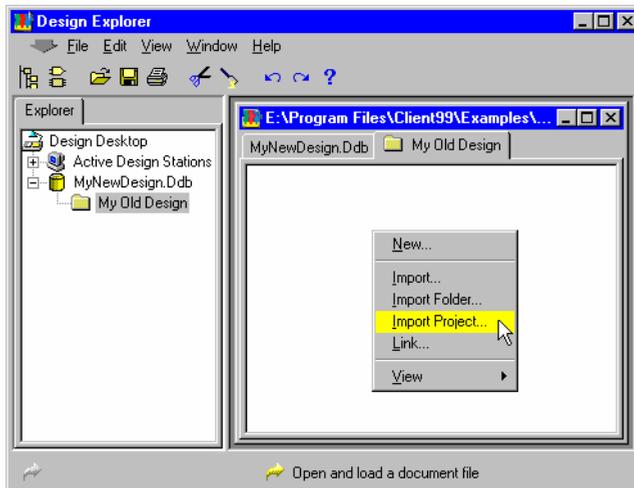


Рис. 2.20. Для импорта старого проекта в новую базу данных проекта необходимо щелкнуть правой кнопкой мыши

Импорт документов с помощью технологии drag-and-drop

Документ может быть импортирован в базу данных проекта непосредственно из Проводника системы Windows с помощью технологии drag-and-drop (захвати и брось). Этот метод работы с документами является наиболее эффективным. Если с помощью мыши захватить и бросить в среду проектирования папку из Проводника, то она будет импортирована в указанную или новую базу данных проекта с сохранением всей структуры вложенных папок и документов (рис. 2.21). Вся иерархическая структура папок будет полностью воспроизведена в проекте.

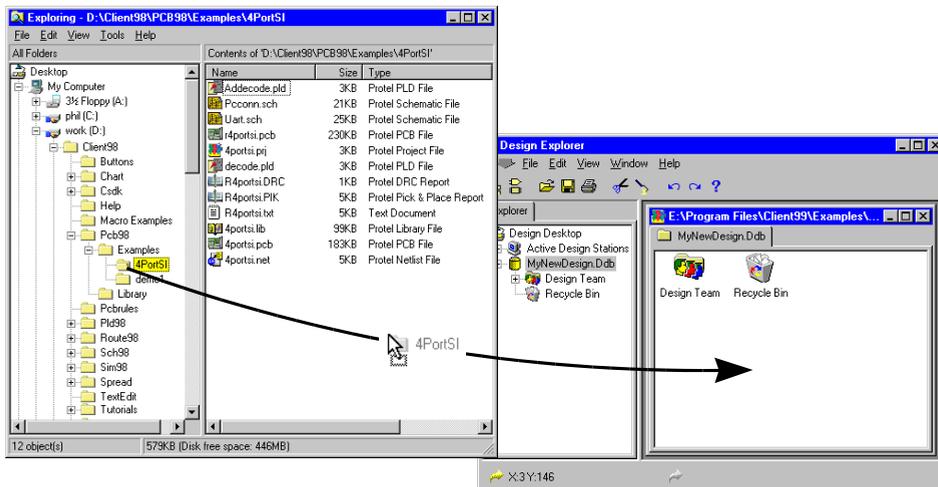


Рис. 2.21. Импорт проекта может быть осуществлен простым перетаскиванием папки старого проекта из Проводника системы Windows в открытую в Protel базу данных проекта

Связывание с внешними документами

Иногда пользователю может понадобиться хранить документ не внутри базы данных проекта, а отдельно на диске. В этом случае будет полезнее не импортировать его, а вставить в базу данных проекта связь с этим документом, при этом он остается на диске компьютера. Это означает, что он может быть использован другими разработчиками, использующими старые версии системы Protel.

Работа с существующими библиотеками элементов

Система Protel 99 SE допускает использование библиотек элементов в виде отдельных файлов с расширением **.LIB**. Самый простой способ использования одних и тех же библиотек разными версиями системы состоит в том, чтобы создать библиотечную базу данных и вставить в нее связи с внешними библиотеками в формате файлов с расширением **.LIB**. В этом случае библиотечная база данных сможет быть использована системой Protel 99 SE, а отдельные файлы — системой Protel 98.

Синхронизация ранее созданных проектов схем и печатных плат

В системе Protel 99 SE информация между листом принципиальной схемы и чертежом печатной платы передается через специальный модуль синхронизации **Design**

Synchronizer. Наличие этого модуля значительно облегчает передачу проектной информации из редактора принципиальных схем в редактор печатных плат и обратно.

Когда пользователь собирается использовать синхронизатор, он сначала должен задать направление передачи информации, для чего надо выбрать из меню команды **Update PCB** (обновить плату) или **Update Schematic** (обновить схему). Далее появится диалоговое окно **Update**, которое будет управлять процессом синхронизации проекта.

Выбор направления синхронизации

Перед тем, как пользователь запустит синхронизатор, он должен выбрать и задать направление синхронизации, то есть определить, что есть источник, а что приемник информации. Проектная информация получается из источника и передается в приемник. В системе Protel 99 SE может быть выполнено обновление чертежа печатной платы по информации из принципиальной схемы, наоборот, обновление принципиальной схемы, исходя из изменений в печатной плате. Так как при сравнении схемы и платы синхронизатор опирается на позиционные обозначения, необходимо следить за их правильным соответствием.

Разумеется, что направление синхронизации пользователь должен выбирать опираясь на свои последние действия с проектом: если он менял схему, то обновлять надо печатную плату командой **Design » Update PCB**, и наоборот, если он добавлял компоненты на плату, то обновлять надо схему командой **Design » Update Schematic**.

Выполнение первичной синхронизации

Для отображения синхронизации элементов принципиальной схемы компонентам на печатной плате синхронизатор проекта использует специальные идентификаторы соответствия. Их присвоение будет выполнено при первичной синхронизации только что импортированных проектов принципиальной схемы и печатной платы.

При первом запуске на выполнение одной из команд **Update** появится одноименное диалоговое окно со списком доступных для обновления документов. После того, как пользователь выберет нужный документ, нажмет кнопку **View Changes** (просмотреть изменения) и запустит процесс нажатием кнопки **Execute**, появится окно **Confirm Component Associations**, которое управляет процессом первичной синхронизации.

Более подробно работа с синхронизатором проекта и диалоговым окном **Confirm Component Associations** описана в разделе *Передача информации о схеме в редактор печатных плат* главы *Разработка принципиальных схем*.

Технологии, используемые внутри системы Protel 99 SE

Система Protel 99 SE построена на базе интегрированной технологии Smart, разработанной компанией Protel — SmartTool, SmartDoc и SmartTeam. Использование этих трех технологий дает пользователю следующие преимущества:

- Разработчик получает законченный инструмент проектирования печатных плат, полностью интегрированный в единый пользовательский интерфейс.
- Все документы проекта хранятся в одной базе данных проекта.
- Широкий набор функций управления групповой соразработкой проекта дает возможность значительно повысить эффективность работы.

Технология SmartTool

Главной частью программной архитектуры системы Protel 99 SE является технология SmartTool, которая в рамках единой среды проектирования Design Explorer объединяет различные редакторы документов внутренних форматов, а также внешние редакторы документов, поддерживающие технологию OLE.

Основу этой технологии составляет архитектура клиент-сервер (рис. 2.22), которая четко разграничивает пользовательский интерфейс (клиент) и инструментарий (серверы).

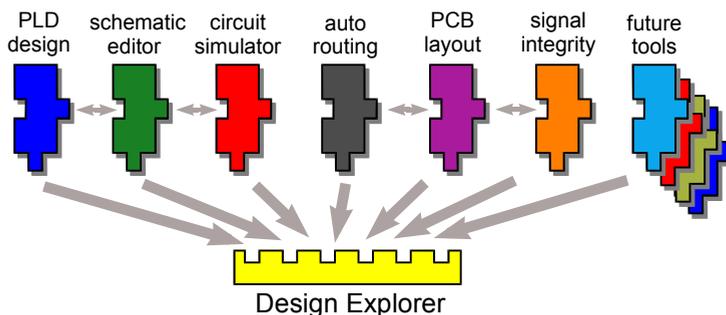


Рис. 2.22. Все серверы встраиваются в оболочку Design Explorer, но при этом могут напрямую связываться друг с другом

Что такое архитектура клиент-сервер?

Архитектура клиент-сервер системы Protel 99 SE представляет собой новый подход к построению систем проектирования. Так как различные инструменты проектирования, как правило, разрабатываются как отдельные автономные приложения, их по-

лезно интегрировать в единый пользовательский интерфейс (клиентское приложение) как модули (серверы). В системе Protel 99 SE клиентским приложением является среда Design Explorer, а различные редакторы и утилиты — серверами.

Что такое Design Explorer?

С программной точки зрения среда Design Explorer является обычным исполняемым приложением, которое запускается из меню Пуск системы Windows как и любая другая программа.

Снаружи среда Design Explorer выглядит как оболочка, предоставляющая пользователю все необходимое для эффективной работы: панель просмотра дерева проекта, меню команд, панели инструментов, горячие клавиши.

Изнутри среда Design Explorer представляет собой мощную программную платформу, обеспечивающую четкое взаимодействие подключенных к ней служебных программ — серверов. Когда к этой платформе подключается очередной сервер, он сообщает системе какие функции (процедуры) он выполняет и передает ей определение всех своих ресурсов: меню команд, панелей инструментов и горячих клавиш. Таким образом, когда пользователь запускает на выполнение ту или иную команду, среда вызывает соответствующий сервер и говорит ему, какую процедуру необходимо выполнить.

Что такое сервер?

Пользуясь языком программистов можно определить сервер как программный модуль, встраиваемый в оболочку Design Explorer и добавляющий в системы новые функциональные возможности. Диапазон этих возможностей может быть очень широким: от законченного редактора документов, например, редактора принципиальных схем Schematic Editor, сложного вычислительного ядра анализа, такого как программа моделирования Circuit Simulator до простой утилиты подсчета количества отверстий в печатной плате.

В системе Protel 99 SE сервера оформлены в виде динамически подключаемых библиотек DLL. В операционной системе Microsoft Windows библиотеки DLL представляют собой внешние функции и процедуры, которые могут быть использованы любым приложением или другой динамической библиотекой. Компания Microsoft разработала программную модель EXE/DLL таким образом, что стало возможным многократное использование программ в одной операционной системе. Программные функции, которые могут быть использованы более чем одним приложением, хранятся в библиотеках DLL и могут вызываться по мере необходимости. Сама же система Windows построена таким образом, что использование таких функций ни скоростью, ни удобством не отличается от использования внутренних функций приложения.

Система Protel 99 SE расширяет эту модель, делая функции и процедуры, сохраненные в серверах DLL, напрямую доступными пользователю через меню, панели инструментов и горячие клавиши.

Аналогично тому, как сервер показывает пользователю содержащиеся в нем функции посредством команд меню, кнопок панелей инструментов и горячих клавиш, он раскрывает их другим серверам через открытый прикладной интерфейс API (Application

Programming Interface), который представляет собой описание, каким образом можно использовать те или иные функции. Открытым такой интерфейс называется в случае, когда все описания опубликованы, и любое приложение (EXE или DLL) может получить доступ к функциям, содержащимся в данной библиотеке.

Аналогично тому, как пользователь получает программный доступ к функциям сервера через ресурсы среды проектирования Design Explorer, интерфейс API предоставляет системе еще более мощные функции непосредственного манипулирования информацией, содержащейся в открытом в данный момент времени документе проекта. Наиболее наглядным здесь является PCB сервер, который при наличии открытого документа в редакторе печатных плат дает возможность автотрассировщику проверить содержимое этого документа через интерфейс API. Пользуясь этим механизмом автотрассировщик получает информацию об объектах на печатной плате и возвращать информацию о расположении проводников и переходных отверстий.

Типы серверов

Все серверы системы Protel 99 SE разделены на следующие категории:

Редакторы или браузеры документов. Эти серверы предоставляют пользователю возможность редактировать или просматривать те или иные документы. Примером являются редакторы принципиальных схем **Schematic Editor** и чертежей печатных плат **PCB Layout Editor**.

Мастера. Такие серверы появляются перед пользователем в виде диалоговых окон, которые через наборы вопросов и ответов дают ему возможность пошагового выполнения тех или иных операции, например, мастер создания топологических посадочных мест компонентов **PCB Component Creation Wizard**.

Утилиты. Эти серверы работают совместно с редакторами документов. Обычно их присутствие выражается в добавлении к командному меню редактора наборов новых команд, расширяющих его функциональные возможности. В качестве примера такого сервера можно привести автотрассировщик печатных плат **PCB Autorouter**, который добавляет в командное меню редактора печатных плат меню **AutoRoute**. Другим примером является программа моделирования **Circuit Simulator**, которая добавляет к меню редактора принципиальных схем команду **Simulate**.

OLE серверы. Данный тип серверов обеспечивают связь документов, сохраненных в базе данных проекта, с внешними приложениями Windows. Например, с помощью такого сервера осуществляется редактирование документов MS Word.

Следует отметить, что существует еще один тип серверов, не выполняющих никаких функций, но содержащих собственные настройки ресурсов, что позволяет автоматически изменять интерфейс среды Design Explorer. Это может быть полезно, например, при локализации интерфейса (переводе основных команд на русский язык).

Более подробная информация о подключении, отключении и запуске тех или иных серверов представлена в подразделе *Работа с серверами* раздела *Изменение настроек среды проектирования Design Explorer*.

Что такое OLE сервер?

Технология OLE (Object Linking and Embedding — встраивание и связывание объектов) представляет собой высокоуровневую технологию обмена данными между приложениями, разработанную компанией Microsoft. Модель OLE состоит из двух отдельных частей: OLE серверов, которые переносят свои функции в другие приложения, и OLE контроллеры, которые обращаются к процедурам внутри OLE серверов. В системе Protel 99 SE роль OLE контроллера выполняет оболочка Design Explorer.

В число приложений, запускаемых как OLE серверы, входят программы Microsoft Word and Excel, а также графический пакет Visio и программы моделирования. Пользователю нет необходимости устанавливать OLE серверы в среду проектирования Design Explorer. Эта оболочка автоматически распознает OLE серверы, установленные на вашем компьютере.

Технология SmartDoc

Технология **SmartDoc** переопределяет понятия интеграции и управления документами проекта, объединяя все документы в единую базу данных проекта **Design Database**. Внутри этой базы данных могут быть сохранены документы самых различных типов: листы принципиальных схем, чертежи печатных плат, выходные файлы в формате Gerber, текстовые документы Microsoft Word и электронные таблицы Microsoft Excel, чертежи, созданные в системе AutoCAD, и рисунки, подготовленные системой Visio.

Внутри среды проектирования пользователь может работать также, как он всегда работал в Проводнике системы Windows. Он может создавать папки и документы, вырезать, копировать, вставлять и удалять их, использовать технологию drag-and-drop (перетаски и брось) для перемещения документов как между папками внутри среды Design Explorer, так и между ней и Проводником системы Windows.

Технология SmartTeam

Технология **SmartTeam** обеспечивает новый подход к совместной работе над проектом группы разработчиков. Она дает возможность одновременного доступа к документам одного и того же проекта всем членам группы. Администратор при этом имеет все необходимые инструменты и функции для управления доступом различных членов группы разработчиков к тем или иным частям проекта. Он может создавать новых членов группы, определять их права, отслеживать, кто из разработчиков и над каким документом работает в настоящий момент, блокировать документы от несанкционированного изменения.

Просматривать, кто и над чем работает, а также блокировать документы от неосторожного изменения может каждый член группы разработчиков.

Управление всеми этими функциями осуществляется только средствами среды Design Explorer на уровне конкретной базы данных проекта, без привлечения средств сетевого управления и разграничения доступа.

Изменение настроек среды проектирования Design Explorer

Среда проектирования Design Explorer является полностью настраиваемой оболочкой, допускающей перемещение и переконфигурацию всех существующих, а также создание новых панелей инструментов, меню команд и горячих клавиш.

Если пользователь хочет создать новую панель инструментов, добавить дополнительный пункт в меню команд, определить новую горячую клавишу, то он прежде всего должен осознать смысл понятий ресурсы и процедуры.

Ресурсы — панели инструментов, меню команд и горячие клавиши

Внутри среды проектирования Design Explorer пользователь может производить различные операции, например, открытие и закрытие документов, их редактирование, генерацию вторичных документов на их основе. Все эти операции инициируются в рамках среды проектирования с помощью меню команд, панелей инструментов и горячих клавиш. Все эти механизмы попадают под понятие ресурсов среды проектирования Design Explorer.

Все ресурсы среды проектирования являются полностью настраиваемыми. Пользователь может добавить кнопку на панель инструментов, реорганизовать меню команд и задать собственные горячие клавиши в любой момент времени.

Понятие ресурсов

Все производимые в рамках среды проектирования Design Explorer операции, например, размещение компонентов, изменение масштаба просмотра, обновление содержимого экрана, выполняются соответствующими процедурами. Доступ к этим процедурам осуществляется с помощью ресурсов. Более подробная информация о процедурах будет представлена в следующем подразделе *Понятие процедуры* данной главы.

Всего в системе Protel 99 SE имеются три типа ресурсов: меню команд, панели инструментов и горячие клавиши. При первоначальной установке их конфигурация задается по умолчанию и сохраняется в специальном общем файле определения ресурсов (resource definition file). Любые изменения настроек, сделанные пользователем, сохраняются в этом файле. В любой момент времени пользователь может вернуться к установкам по умолчанию.

Меню команд

Все меню команд системы Protel 99 SE организованы по образцу и подобию любых других приложений системы Windows. Это означает, что все стандартные операции, такие как открытие и закрытие файлов, вырезание или вставка объектов, вывод до-

кументов на печать, будут выполняться одинаково во всех редакторах системы. Такой подход делает работу в разных редакторах системы чрезвычайно эффективной.

Для редактирования меню команд выполните на нем двойной щелчок левой кнопкой мыши.

Всплывающие меню команд

В среде Design Explorer имеются специальные горячие клавиши, предназначенные для быстрого вызова всплывающих меню во время работы с каким либо графическим редактором, когда использование мыши невозможно или затруднено. Например, при нажатии клавиши **F** в редакторе принципиальных схем появляется всплывающее меню команд **File**, при нажатии клавиши **T** — меню команд **Tools**. Такой подход позволяет значительно ускорить работу разработчика, за счет быстрого и удобного доступа к командам меню и подменю через клавиатуру.

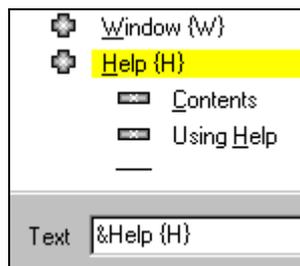


Рис. 2.23. Задание горячих клавиш вызова всплывающих меню.

Горячая клавиша вызова всплывающего меню задается при редактировании меню команд в специальном текстовом поле как буква в фигурных скобках { } (рис. 2.23).

Панели инструментов

Панели инструментов могут быть прикрепленными к любой стороне оболочки Design Explorer или располагаться в любом удобном месте.

Можно без труда изменить существующую или создать новую панель инструментов, связав с кнопками любую из доступных в настоящий момент процедур. Пример создания пользовательской панели инструментов будет приведен несколько позже.

Для редактирования панели инструментов выполните двойной щелчок левой кнопкой мыши на ее названии.

Горячие клавиши

Каждый редактор документов системы Protel 99 SE имеет как минимум один набор горячих клавиш. Разумеется, этот набор может быть отредактирован, и в него могут быть добавлены новые горячие клавиши.

Горячие клавиши могут быть заданы как одиночные, так и как комбинации нескольких клавиш с использованием функциональных клавиш **CTRL**, **SHIFT** и **ALT**.

При назначении горячих клавиш, следует помнить, что они могут совпасть с уже существующими настройками всплывающих меню. В этом случае настройки горячих клавиш будут иметь больший приоритет, чем всплывающие меню.

Установки меню команд, панелей инструментов и горячих клавиш по умолчанию

Установки ресурсов каждого сервера по умолчанию определяются в специальном файле с расширением **.RCS**. Этот файл содержит установки по умолчанию для всех без исключения ресурсов системы. Ни один из этих ресурсов не может быть удален из системы, но он может быть легко перенастроен.

В процессе инсталляции системы Protel 99 SE настройки по умолчанию для всех модулей программы загружаются в среду проектирования Design Explorer и сохраняются в общем файле описания ресурсов. Пользователь может изменить и сохранить настройки в любой момент работы над проектом.

Сброс всех пользовательских настроек

Пользователь может на свой вкус перенастраивать любые меню команд, панели инструментов и горячие клавиши без каких-либо ограничений. Тем не менее, всегда может появиться необходимость вернуться к настройкам ресурсов по умолчанию, что пользователь может сделать в любой момент работы над проектом, выполнив команду меню **Client Menu » Servers**. Далее необходимо выполнить двойной щелчок левой кнопкой мыши на иконке нужного сервера и в появившемся диалоговом окне **EDA Servers** нажать кнопку **Defaults**. Все пользовательские настройки сервера будут сброшены и возвращены в состояние по умолчанию.

Иногда после выполнения этой операции пользователь может обнаружить, некоторые из ранее существовавших команд меню исчезли. Это случилось из-за того, что данные команды были введены в меню другими серверами. Например, после сброса пользовательских настроек и восстановления настроек ресурсов по умолчанию для редактора принципиальных схем **Schematic Editor** из меню команд исчезнут группы команд **Simulate** и **PLD**. Восстановить их можно сбросив настройки для серверов **Sim server** и **PLD server**.

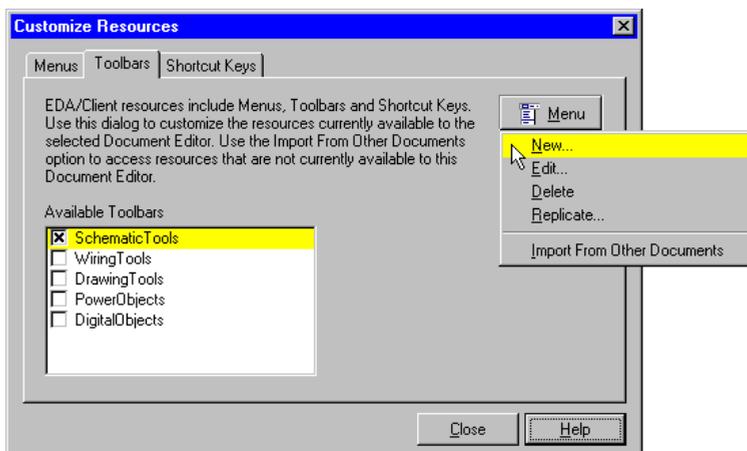


Рис. 2.24. Настройка ресурсов среды Design Explorer.

Изменение настроек ресурсов

Процесс создания пользовательской горячей клавиши, изменения состава меню команд или панели инструментов представляет собой не что иное как изменение настроек ресурсов.

Для того, чтобы изменить настройки ресурсов для активного редактора документов, необходимо выполнить команду меню **Client menu » Customize**, после чего появится диалоговое окно **Customize Resources** (рис. 2.24). При активном листе принципиальной схемы это окно будет содержать список всех ресурсов, доступных редактору принципиальных схем **Schematic Editor**.

Выбрать нужный тип ресурсов можно щелкнув на соответствующей вкладке, а нажав на кнопку **Menu** можно вызвать список команд работы с ресурсами.

С помощью приведенных здесь команд пользователь может:

Удалить ресурс (Delete)

Когда выбирается данная команда система попросит подтвердить удаление выбранного ресурса. Здесь же будет находиться опция **Remove from Global Resource Pool**, которая разрешает полное удаление выбранного ресурса из системы. Если эта опция останется выключенной, то будет удалена только связь этого ресурса с соответствующим редактором, а сам он из среды проектирования Design Explorer удален не будет.

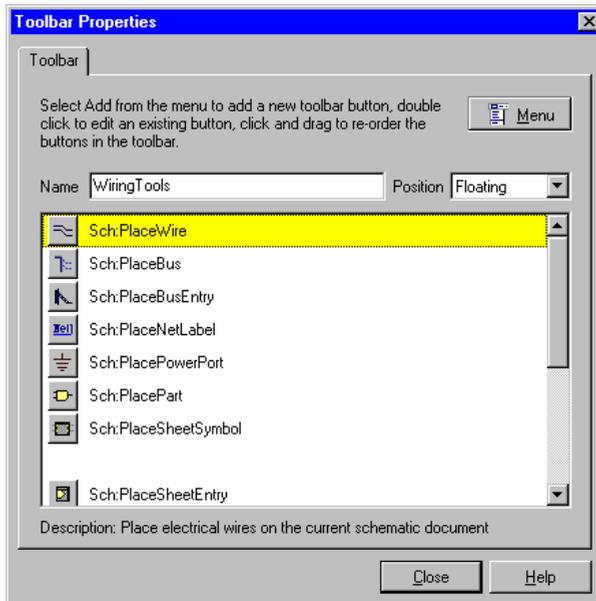


Рис. 2.25. Настройка панели инструментов.

Дублировать ресурс (Replicate)

Эта команда выбирается в случае, если пользователь хочет создать новый ресурс не "с нуля", а лишь несколько видоизменив уже существующий.

Импортировать ресурс из других документов (Import from Other Documents)

Эта команда используется, когда необходимо получить доступ к ресурсам, принадлежащим другим серверам или ранее созданным пользовательским ресурсам.

Редактировать ресурс (Edit)

Данная команда вызывает семейство диалоговых окон, позволяющих редактировать существующие ресурсы: меню команд, панели инструментов и горячие клавиши (рис. 2.25).

◆ Ряд горячих клавиш существует и для работы с диалоговыми окнами редактирования ресурсов. Здесь работают: контекстное меню вызываемое нажатием правой клавиши мыши, клавиши **INSERT** и **DELETE**, используется технология drag-and drop.

Назначение выбранному ресурсу нужной процедуры

Для построения нового ресурса необходимо выполнить два обязательных действия:

1. Создать новый ресурс (кнопку на панели инструментов, команду меню или горячую клавишу).
2. Назначить процедуру, которая будет выполняться при обращении к данному ресурсу.

Процесс назначения процедуры не зависит от типа редактируемого ресурса. В любом случае появится диалоговое окно, в котором надо задать имя и местонахождение нужной процедуры (рис. 2.26).



Рис. 2.26. Назначение процедуры редактируемому ресурсу.

Поиск процедуры

В поле **Process** задается имя процедуры, которая будет запускаться при обращении к данному ресурсу. Для поиска нужной процедуры необходимо нажать кнопку **Browse**, после чего появится диалоговое окно Process Browser где с помощью фильтра можно отобрать и найти нужную процедуру из всех доступных системе.

Получение описания процедуры

Чтобы получить интерактивную справку с описанием выбранной процедуры, необходимо нажать кнопку **Info**, расположенную в окне **Process Browser**.

Пример создания новой панели инструментов

Имеется два способа создания новой панели инструментов: продублировать и видоизменить существующую панель инструментов или создать ее заново. Приведенный ниже пример показывает процесс создания "с нуля" новой панели инструментов размещения графических примитивов в редакторе чертежей печатных плат **PCB Editor**.

1. Сделайте активным любой чертеж печатной платы и выполните команду меню **Client menu » Customize**.
2. Щелкните в появившемся диалоговом окне левой кнопкой мыши на вкладке **Toolbar** и выберите команду **New** во всплывающем меню, вызываемом нажатием кнопки меню, после чего появится окно **Toolbar Properties**.
3. Впишите в поле **Name** имя новой панели инструментов "**Primitives**".
4. Выполните один щелчок левой кнопкой мыши в пустом белом поле в окне и нажмите клавишу **INSERT**. В поле появится новая кнопка, которой надо назначить нужную процедуру.
5. Выполните двойной щелчок левой кнопкой мыши на этой кнопке, в результате чего появится диалоговое окно **Button**.
6. Нажмите кнопку **Browse**, после чего появится диалоговое окно **Process Browser**.

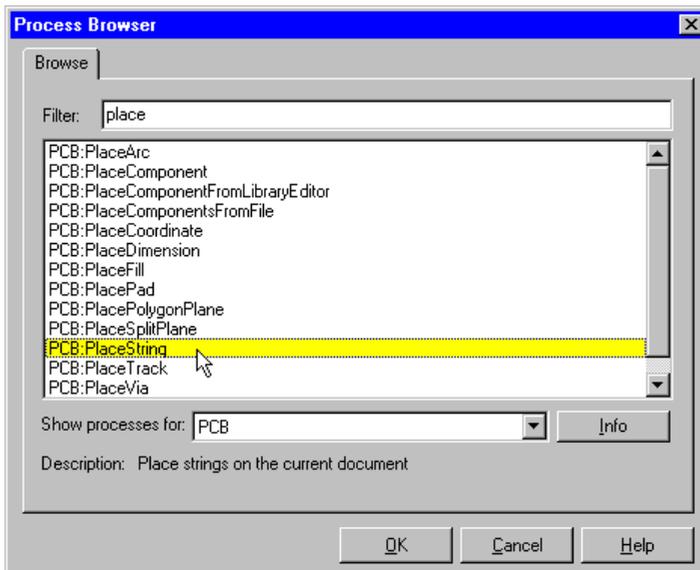


Рис. 2.27. Использование фильтра для выбора процедуры.

7. В выпадающем списке **Show Processes** for выберите строку **PCB**.
8. В поле **Filter** напишите слово **"place"**.

После этого в списке доступных процедур будут отображены все процедуры PCB сервера, начинающиеся со слова **place** (рис. 2.27). Обратите внимание, что имя сервера, расположенное слева от имени процедуры и отделенное от него двоеточием, фильтром не рассматривается. Кроме того, в фильтре можно использовать групповые символы подмены * (набор любых символов) и ? (один любой символ).

9. Выберите процедуру **PCB:PlaceString** и нажмите кнопку **OK** или выполните двойной щелчок левой кнопкой мыши на этой процедуре.
10. Для обозначения кнопки мы будем использовать уже существующую картинку, адрес которой на диске необходимо указать в поле **Bitmap File**. Для этого нажмем кнопку **Browse**, расположенную рядом с этим полем, после чего появится диалоговое окно выбора файла **Image File**. Все картинки расположены в папке **\Program Files\Design Explorer 99 SE\System\Buttons**.
11. В поле **File Name** введите строку **"t*.bmp"**. В окне отобразятся все файлы, имя которых начинается с буквы **"t"**.
12. Выполните двойной щелчок левой кнопкой мыши на файле **TEXT.BMP**. Диалоговое окно **Image File** закроется и мы вернемся в окно **Button**. Обратите внимание, в нижней части окна экрана появилась кнопка с большой буквой **T** (рис. 2.28).

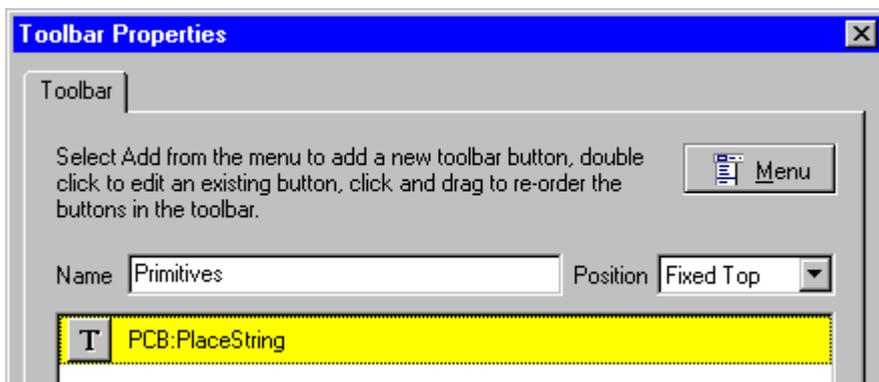


Рис. 2.28. Первая кнопка на новой панели инструментов Primitives

13. Нажмите кнопку **OK**, после чего диалоговое окно **Button** закроется.
14. Повторите шаги 4 — 13 и добавьте на панель еще пять кнопок для процедур, приведенных в таблице 2.2.

Таблица 2.2

Имя процедуры	Картинка
PCB:PlaceArc	arc.bmp
PCB:PlaceFill	rect.bmp
PCB:PlaceVia	via.bmp
PCB:PlacePad	pad.bmp
PCB:PlaceTrack	track.bmp

15. Все шесть кнопок будут располагаться на панели в один ряд. Чтобы расположить кнопки в два ряда по три штуки, выделите четвертую кнопку в списке и выполните команду **Separator** из списка команд, появляющегося при нажатии на кнопку **Menu**. В списке кнопок появится пустая строка.

16. Чтобы закрыть окно **Toolbar Properties**, нажмите кнопку **Close**.

Будет создана новая панель инструментов, имя которой появится в списке доступных панелей инструментов в диалоговом окне **Customize Resources**.

17. Чтобы закрыть окно **Customize Resources**, нажмите кнопку **OK**.

Теперь в списке ресурсов среды проектирования будет присутствовать новая панель инструментов **Primitives** (рис. 2.29), которая будет автоматически связана с редактором чертежей печатных плат.

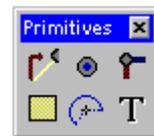


Рис. 2.29. Новая панель инструментов.

Понятие процедуры

Функциональные возможности каждого сервера, подключенного к оболочке Design Explorer, реализуются посредством выполнения встроенных в него процедур. Понимание того, как они запускаются и работают позволит пользователю работать в среде проектирования намного эффективнее.

Что такое процедура?

Понятие процедуры в системе Protel 99 SE ничем не отличается от такового в программировании — это строго заданная последовательность выполняемых программой операций. Некоторые из таких операций могут быть простыми, например, обновление экрана, другие — достаточно сложными, например, прорисовка полигонов.

Любая операция, выполняемая в среде проектирования Design Explorer, управляется определенной процедурой. Если пользователь выполняет команду меню **File » Save**, запускается процедура **SaveDocument**. Выполнение команды меню **Place » Wire**, запускает процедуру **PlaceWire** модуля **Advanced Schematic**, который дает возможность пользователю интерактивно размещать проводники на листе принципиальной схемы. Выполнение любой команды меню, нажатие кнопки на панели инструментов или горячей клавиши вызывают соответствующую подпрограмму сервера, поэтому их называют пускателями процедур (Process Launcher). Все же выполняемые после этого операции называют процедурой.

Каждая процедура имеет уникальное имя — идентификатор процедуры (Process Identifier), который содержит имя сервера и процедуры внутри него. Ниже приведены два примера записи идентификатора процедур.

Client:SaveDocument

Sch:PlaceWire

В справочной системе среды проектирования Design Explorer приведено подробное описание всех существующих процессов. Чтобы вызвать соответствующий раздел справки, необходимо нажать кнопку **Info**, присутствующую во всех диалоговых окнах, связанных с обработкой процедур.

Запуск процедуры

Когда пользователь выбирает какой-либо пункт меню или нажимает кнопку на панели инструментов, он запускает некоторую процедуру. Запуск процедуры осуществляется посредством передачи серверу соответствующего идентификатора процедуры, после чего сервер автоматически начинает ее выполнение.

Для передачи серверу идентификатора процедуры используются так называемые пускатели процедур (Process Launcher): команды меню, кнопки на панелях инструментов и горячие клавиши (рис. 2.30).

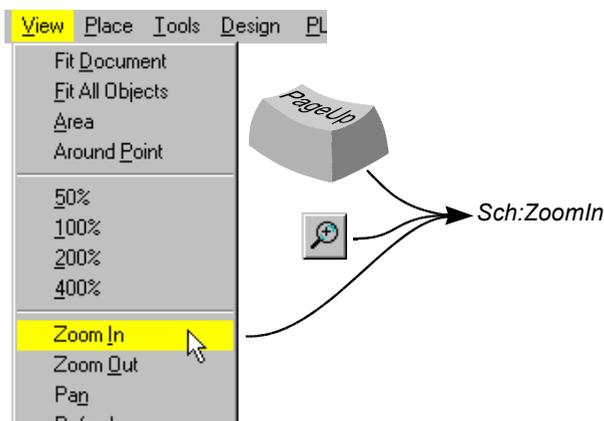


Рис. 2.30. Для запуска процедуры Sch:ZoomIn могут быть использованы три типа пускателей: кнопка на панели инструментов, команда меню и горячая клавиша

Параметры процедур

Все процедуры, выполняемые в среде проектирования, являются параметрическими. Это означает, что в сервер передается не только идентификатор процедуры, но и некоторый набор параметров, определяющий инструкции по выполнению представленных в процедуре действий. То есть, все происходит аналогично тому, как это делается при вызове подпрограммы в программировании.

Рассмотрим процедуру **Sch:PlacePart**. Когда пользователь запускает эту процедуру, например, выбирая команду меню **Place » Part**, появляется диалоговое окно, предлагающее задать имя элемента библиотеки. После ввода имени, появится окно, запрашивающее ввод позиционного обозначения, и лишь после этого выбранный элемент появится в поле редактора схем "приклеенным" к указателю мыши. Вместо того, чтобы указывать имя и позиционное обозначение элемента в упомянутых диалоговых окнах, пользователь может передать серверу эту информацию при вызове процедуры в виде списка параметров.

Для передачи серверу списка параметров надо войти в режим редактирования кнопки и перечислить их в поле **Parameters**, расположенном окне в окне **Button** (рис. 2.31).

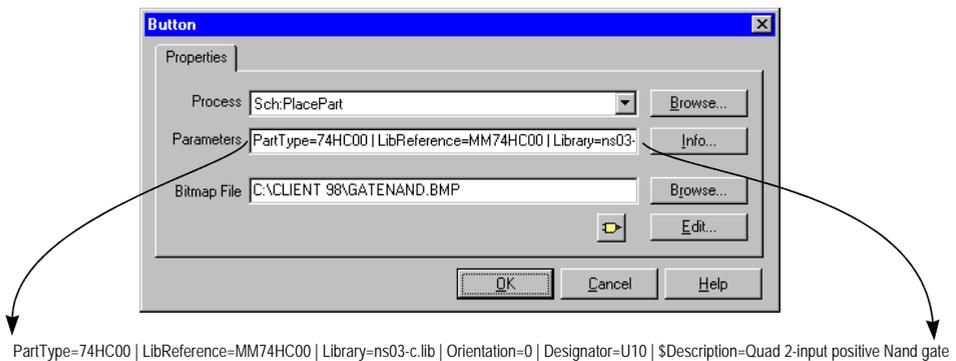


Рис. 2.31. Редактирование кнопки на панели инструментов, запускающей процедуру Sch:PlacePart. В процедуру также передается набор параметров

Library	= имя библиотеки, содержащей нужный компонент. Если библиотека располагается не в стандартной папке, ее адрес требуется указать полностью.
LibReference	= имя компонента в библиотеке.
Designator	= присваиваемое компоненту позиционное обозначение.
PartType	= тип или описание компонента.
Orientation	= ориентация компонента, относительно библиотечной.
\$Description	= текстовая подсказка, появляющаяся при наведении указателя мыши на данную кнопку.

После этого нажатие на данную кнопку будет вызывать процедуру размещения элемента, причем "приклеенной" к курсору окажется именно микросхема 74НС00 с уже назначенным позиционным обозначением.

◆ Информацию о наборе параметров той или иной процедуры можно получить из справочной системы Protel 99 SE нажав на кнопку Info.

Правила записи списка параметров

Ниже приведен общий вид записи списка параметров процедуры:

```
parameter1 = value1 | parameter2 = value2 | parameter3 = value3
```

Порядок следования параметров в списке произвольный, разделителем является символ вертикальной черты "|". Имя параметра может быть указано в как прописными, так и строчными буквами, то есть система не чувствительна к регистру.

Подробную информацию о наборе параметров различных процедур можно получить из интерактивной справочной системы Protel 99 SE, нажав кнопку Info в диалоговых окна редактирования ресурсов Menu, Button или Keyboard.

Работа с серверами

При инсталляции системы Protel 99 SE все серверы будут автоматически установлены и подключены к оболочке Design Explorer. В общем случае, пользователю нет необходимости дополнительно вручную устанавливать или удалять отдельные серверы. Так как все серверы подключаются к системе по мере необходимости и не загружают память компьютера, у пользователя нет причин удалять те или иные неиспользуемые серверы из среды проектирования Design Explorer.

Типы серверов

Среда проектирования Design Explorer поддерживает четыре основных типа серверов:

- Protel Document Editors — редакторы стандартных документов системы Protel, например, Schematic Editor.
- Protel Wizards — мастера, направляющие действия пользователя при выполнении сложных операций, например, мастер создания печатных плат PCB Board Creation Wizard.
- Protel Utility Servers — утилиты, такие как автотрассировщик PCB Autorouter.
- OLE Servers — серверы, поддерживающую технологию OLE, например, редактор Word.

Система Protel 99 SE объединяет исчерпывающий набор редакторов стандартных документов:

Schematic Editor — редактор листов и элементов принципиальных схем.

PCB Editor — редактор чертежей печатных плат и топологических посадочных мест компонентов.

Text Editor — редактор текстовых документов, таких как списки соединений и исходные файлы для проектов на базе ПЛИС.

Spread — редактор электронных таблиц.

Chart — редактор диаграмм на базе данных из электронных таблиц.

Macro — редактор макросов, предоставляющий пользователю базовые возможности программирования.

Pro Здесь также содержится большое число утилит:

Circuit Simulator — программа моделирования аналоговых и цифровых схем.

Netlister — утилита генерации списков соединений в различных форматах.

Design Synchronizer — синхронизатор проекта, обеспечивающий согласованность данных на принципиальных схемах и печатных платах.

PLD Compiler — компилятор проектов на базе ПЛИС, описанных принципиальной схемой или на языке CUPL-HDL.

PCB Autoplacer — модуль автоматического размещения компонентов на печатных платах.

PCB Manual Router — сервисная утилита, обеспечивающая все функции интерактивной трассировки.

PCB Autorouter — автотрассировщик печатных плат.

Большое количество вспомогательных утилит, позволяющих создавать списки используемых материалов (Bill Of Material, BOM), библиотеки, экспортировать данные в электронные таблицы и многое другое.

Инсталляция сервера

При инсталляции системы Protel 99 SE со стандартного компакт-диска все серверы будут установлены автоматически. Если пользователю все же потребовалось установить какой либо сервер отдельно, он должен выполнить следующую последовательность действий.

1. Выполнить команду меню **Client menu » Servers**. Появится окно **EDA Servers**, в котором показаны все установленные в настоящий момент серверы.
2. Нажать кнопку **Menu** и в появившемся списке выбрать команду **Install** (рис. 2.32). Появится диалоговое окно открытия документа.
3. Выбрать инсталлятор нужного сервера (файл с расширением **.INS**), например, **ADV SCH.INS**, **SPREAD.INS**.
4. Для запуска инсталляции нажать кнопку **Open**.

В строке **Status**, расположенной в нижней части диалогового окна **EDA Servers**, подтверждается корректность инсталляции выбранного сервера, а также его текущее состояние.



Рис. 2.32. Инсталляция сервера в среду проектирования Design Explorer.

Запуск и останов сервера

Сразу после инсталляции сервер находится в неактивном состоянии, о чем в строке **Status** выдается сообщение **Not Started**. В этом состоянии он не потребляет никаких машинных ресурсов. Запуск сервера произойдет автоматически при первой попытке его использовать, в противном случае его можно запустить вручную. Как правило, пользователю не требуется совершать каких-либо действий для принудительной остановки работы того или иного сервера, за исключением тех случаев, когда требуется высвободить дополнительный объем памяти.

Удаление сервера из системы

В общем случае удаление серверов из системы не требуется, тем не менее, такая возможность существует. Следует помнить, что при этом сервер не удаляется с жесткого диска компьютера, а просто отключается от среды проектирования Design Explorer.

Чтобы удалить сервер из системы требуется:

1. Выполнить команду меню **Client menu** » **Servers**. Появится окно **EDA Servers**, в котором показаны все установленные в настоящий момент серверы.

2. Одним щелчком левой кнопки мыши выбрать из списка нужный сервер.
3. Нажать кнопку **Menu** и в появившемся списке выбрать команду **Remove**.
4. Подтвердите свои намерения, нажав кнопку **Yes** в появившемся окне.

При работе с серверами пользователю следует помнить ряд важных правил.

- Сервер не может быть удален из системы, пока он запущен. Для этого его надо остановить. Но сервер не может быть остановлен, пока существует хотя бы один открытый документ, использующий его.
- Когда пользователь удаляет сервер из среды проектирования Design Explorer, вместе с ним удаляются все соответствующие ресурсы. Все пользовательские установки ресурсов (панели инструментов, меню команд, горячие клавиши) будут также удалены.

Если пользовательских установок было много, их восстановление может оказаться очень трудоемкой задачей, поэтому для этого следует использовать специальные серверы настройки ресурсов. Простейшим примером такого сервера могут служить серверы обновления Protel 99 SE Service Pack, которые запускаются всего один раз и вносят в систему все необходимые изменения.

Создание пользовательских серверов управления ресурсами

Все пользовательские настройки ресурсов носят временный характер и сохраняются в файле **Client99SE.rcs**. На случай сбоя система Protel 99 SE создает резервную копию этого файла под именем **Client99SE.~cs**, однако при восстановлении конфигурации ресурсов по умолчанию все введенные пользователем изменения будут утеряны и операцию настройки придется повторить. Для автоматизации процесса настройки ресурсов рекомендуется использовать специальные серверы, устанавливаемые в систему стандартным образом, но не имеющие собственных процедур, а лишь изменяющие стандартный набор ресурсов.

Для создания такого сервера пользователю достаточно иметь простейший текстовый редактор (например, стандартный Блокнот системы Windows) и простейший графический редактор (например, Paint). С помощью Блокнота создаются два основных элемента сервера: файл инсталлятора и файл ресурсов). Графический редактор используется для создания иконок кнопок пользовательских панелей инструментов.

Правила создания файла ресурсов

В общем случае все элементы серверов среды Design Explorer (за исключением иконок кнопок) находятся в ее системной папке, например, **C:\Program Files\Desing Explorer 99 SE\System**. Графические изображения кнопок должны располагаться в папке **C:\Program Files\Design Explorer 99 SE\System\Buttons**. Однако, это требование не является обязательным, поэтому допускается размещать сервер в произвольной папке на любом из дисков компьютера. Отметим, что в этом случае требования к синтаксису написания файла ресурсов несколько ужесточаются, о чем будет сказано ниже.

Каждое изменение ресурсов должно описываться отдельным текстовым блоком, имеющим следующую структуру:

```
UpdateResource – заголовок блока
Begin – начало
  Kind <тип ресурса>
  Name <имя ресурса>
  Action <действие>
  ReferencePosition <место приложения нового элемента ресурса>
  ReferenceItem <имя существующего элемента ресурса>
  Description – описание изменений
  Begin – начало
    <элемент ресурса> <адрес иконки> <процедура со списком параметров>
    <элемент ресурса> <адрес иконки> <процедура со списком параметров>
    <разделитель> – разделитель, если необходим
    < элемент ресурса > <адрес иконки> <процедура со списком параметров>
    < элемент ресурса > <адрес иконки> <процедура со списком параметров>
  End – конец действия
End – конец действия
```

Здесь ключевое слово <тип ресурса> может принимать одно из трех возможных значений (**ToolBar**, **Menu HotKeyTable**) и описывает тип модифицируемого ресурса. Параметр <имя ресурса> задает имя модифицируемого или создаваемого ресурса (меню команд, панели инструментов или набора горячих клавиш). Точное имя уже существующего ресурса можно получить, открыв текстовым редактором файл **Client99SE.rcs**.

Параметр <действие> описывает режим изменения элементов ресурсов и может иметь значения **Add** (добавить) или **Delete** (удалить). Далее с помощью ключевого слова **ReferencePosition** следует указать куда будет добавляться элемент ресурса (**End** — в конец списка, **Begin** — в начало списка, **Before** — до указанного параметром **Referenceltem** элемента, **After** — после указанного параметром **Referenceltem** элемента). Параметр **Referenceltem** задает имя уже существующего элемента ресурса, которое можно получить из файла **Client99SE.rcs**.

Затем в блоке описания модифицируемых элементов ресурса следует указать имя нового элемента, ссылку на иконку (если речь идет о кнопке на панели инструментов) и идентификатор процедуры с перечнем всех необходимых параметров.

Пример создания пользовательской панели инструментов

Рассмотрим пример создания сервера поддержки **Support**, добавляющего в среду проектирования Design Explorer специальную пользовательскую панель инструментов с именем "**Компоненты по ГОСТ**", которая будет вызываться нажатием специальной кнопки на главной панели инструментов редактора схем, а также специальной командой меню. Данная панель значительно упрощает прорисовку принципиальных схем в системе Protel 99 SE.

Создадим в месте установки системы Protel 99 SE по умолчанию (папка **C:\Program Files\Design Explorer 99 SE**) новую папку **Support**, а в ней два новых текстовых файла **Support.ins** и **Support.rcs**. Создадим в этой папке также необходимый набор иконок в формате BMP размером 18x18 пикселей и палитрой 16 цветов. Проще всего

это сделать скопировав наиболее подходящую иконку из папки Buttons и слегка ее отредактировать. Откроем пустой файл **Support.rcs** с помощью блокнота и начнем описание ресурсов которые должны быть изменены в ходе инсталляции данного сервера.

Сначала опишем саму новую панель инструментов.

```
Toolbar 'Компоненты по ГОСТ' 'Floating' 100 400
Begin

Button 'C:\Program Files\Design Explorer 99 SE\Support\Resistor.bmp'
'Sch:PlacePart (PartType=100k|LibReference=РЕЗИСТОП|Design=\Support\
Support.ddb|Library=\Библиотеки по ГОСТ\Резисторы.lib|Orientation=0|
Designator=R?|Description=Резистор постоянный простой) '

Button 'C:\Program Files\Design Explorer 99 SE\Support\Resistor0050.bmp'
'Sch:PlacePart (PartType=100k|LibReference=РЕЗИСТОП_0_05_ВТ|Design=\
Support\Support.ddb|Library=\Библиотеки по ГОСТ\Резисторы.lib|
Orientation=0|Designator=R?|Description=Резистор постоянный 0.05 Вт) '

Separator

Button 'C:\Program Files\Design Explorer 99 SE\Support\Resistor0125.bmp'
'Sch:PlacePart (PartType=100k|LibReference=РЕЗИСТОП_0_125_ВТ|Design=\
Support\Support.ddb|Library=\Библиотеки по ГОСТ\Резисторы.lib|
Orientation=0|Designator=R?|Description=Резистор постоянный 0.125 Вт) '

Button 'C:\Program Files\Design Explorer 99 SE\Support\Resistor0250.bmp'
'Sch:PlacePart (PartType=100k|LibReference=РЕЗИСТОП_0_25_ВТ|Design=\
Support\Support.ddb|Library=\Библиотеки по ГОСТ\Резисторы.lib|
Orientation=0|Designator=R?|Description=Резистор постоянный 0.25 Вт) '

End
```

Данный блок описывает панель инструментов "Компоненты по ГОСТ", не привязанную ни к одной стороне окна среды проектирования (**Floating**) и расположенную в позиции 100 точек вправо по оси X и 400 вниз точек по оси Y от левого верхнего угла этого окна (рис. 2.33). Все эти данные приведены в первой строке блока. Далее следует описание четырех кнопок, вызывающих процедуру размещения компонента. Рассмотрим описание одного из элементов ресурса более подробно.



Рис. 2.33. Пользовательская панель инструментов "Компоненты по ГОСТ".

- Button — говорит о том, что элементом ресурса является кнопка;
- 'C:\Program Files\Design Explorer 99 SE\Support\Resistor.bmp' — дает полный путь для загрузки иконки для данной кнопки. Если указать только имя файла иконки без полного пути, то система будет пытаться найти иконку в папке **C:\Program Files\Design Explorer 99 SE\System\Buttons**;

Далее следует описание исполняемой процедуры с перечислением параметров в круглых скобках и символом "|" в качестве разделителя.

- Sch:PlacePart — имя процесса **Place Part** из состава сервера **Sch**;
- PartType=100k — номинал резистора по умолчанию;
- LibReference=РЕЗИСТОР — имя компонента в библиотеке;
- Design=\Support\Support.ddb — база данных, содержащая нужную библиотеку;
- Library=\Библиотеки по ГОСТ\Резисторы.lib — имя библиотеки, содержащий нужный компонент;
- Orientation=0 — ориентация компонента в градусах (допускаются значения 0, 90, 180, 270);
- Designator=R? — позиционное обозначение по умолчанию;
- \$Description=Резистор постоянный простой — подсказка, появляющаяся при наведении указателя мыши на данную кнопку;

Аналогичным образом описываются другие кнопки панели, при необходимости с помощью ключевых строк **Separator** добавляются разделители. Завершается блок строкой **End**. Отметим, что в папке с создаваемым сервером должна присутствовать база данных проекта **Support.ddb**, содержащая библиотеку **Резисторы.lib** в папке **Библиотеки по ГОСТ**.



Рис. 2.34. Кнопка включения панели инструментов "Компоненты по ГОСТ".

Далее создадим блок, описывающий добавление на главную панель инструментов редактора схем новой кнопки, включающей или выключающей панель "**Компоненты по ГОСТ**" (рис. 2.34).

```
UpdateResource
Begin
  Kind ToolBar
  Name SchematicTools
  Action Add

  ReferenceItem 'C:\PROGRAM FILES\DESIGN EXPLORER 99
  SE\SYSTEM\BUTTONS\WIRTOOLS.BMP' *

  ReferencePosition After
  Description
  Begin
    Button 'C:\Program Files\Design Explorer 99
    SE\Support\TransistorPNP.bmp' 'Cli-
    ent:CustomizeResources (ToggleToolbar=Компоненты по
    ГОСТ|$Description=Включить/выключить панель Компоненты по ГОСТ) '
  End
End
```

Данный блок предписывает на панель инструментов **SchematicTools** добавить после кнопки с иконкой **WIRTOOLS.BMP** новую кнопку с иконкой, расположенной по адресу **C:\Program Files\Design Explorer 99 SE\Support\TransistorPNP.bmp**. Новая кнопка будет переключать режим отображения панели "**Компоненты по ГОСТ**" (параметр **ToggleToolbar**) с помощью процедуры **CustomizeResources** сервера **Client**.

Включение или выключение панели инструментов можно выполнять с помощью специальной команды меню (рис. 2.35), которую следует описать аналогичным образом.

```
UpdateResource
Begin
  Kind Menu
  Name SchematicMenu
  Action Add
  ReferencePosition After
  ReferenceItem 'Di&gital Objects' *
  Description
  Begin
    MenuItem 'Компоненты по ГОСТ'
    'Client:CustomizeResources (ToggleToolbar=Компоненты по ГОСТ|
    $Description=Включить/выключить панель Компоненты по ГОСТ) '
  End
End
```

Здесь новая команда добавляется в меню редактора схем (параметр **SchematicMenu**) после уже существующей команды **Digital Objects** в меню **View » Toolbars**. Обратите внимание, что здесь приводится имя пункта ресурса "**Di&gital Objects**" в точном соответствии с его описанием в соответствующем сервере, в частности, здесь присутствует символ "**&**", назначающей данной команде горячую клавишу **G** (в меню она будет подчеркнутой).

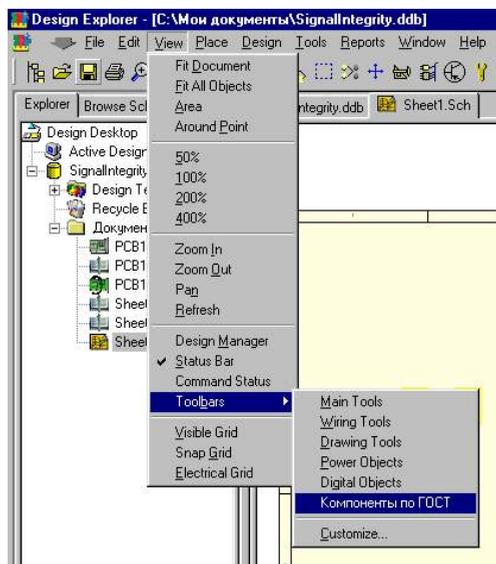


Рис. 2.35. Команда меню включения панели инструментов "Компоненты по ГОСТ".

После сохранения файла **support.rcs** следует подготовить файл инсталляции пользовательского сервера. Для этого с помощью Блокнота откроем файл **support.ins** введем в него следующий текст:

```
ClientInsFile 0.0
EditorExePath 'ADV SCH.DLL'
EditorName 'Support'
EditorDescription      'Поддержка от компании ЭлектронТрейд'
Version                'Версия 1.1.3'
Date                   '1 Сентября 2001 г.'
HelpAboutInfo          'Данное программное обеспечение распространяется в
                        рамках технической поддержки лицензионным пользо-
                        вателям системы Protel 99 SE.'
CopyRight              'Copyright © ElectronTrade 2001, Потапов Ю.В.'
EditorToolbar          'Sch'      'Компоненты по ГОСТ'  'OFF'
```

В строке **EditorExePath** задается имя библиотеки, откуда будет взята иконка для отображения в списке установленных в систему серверов. Строка **EditorName** задает имя сервера. Далее идет список необязательных параметров, которые будут отображаться при вызове справки **About** при нажатии правой кнопки мыши на данном сервере в окне **EDA Servers**. Самая последняя строка файла является очень важной, так как предписывает системе подключить панель инструментов "**Компоненты по ГОСТ**" к редактору схем (параметр **'Sch'**) и по умолчанию она должна быть в выключенном состоянии (параметр **'OFF'**).

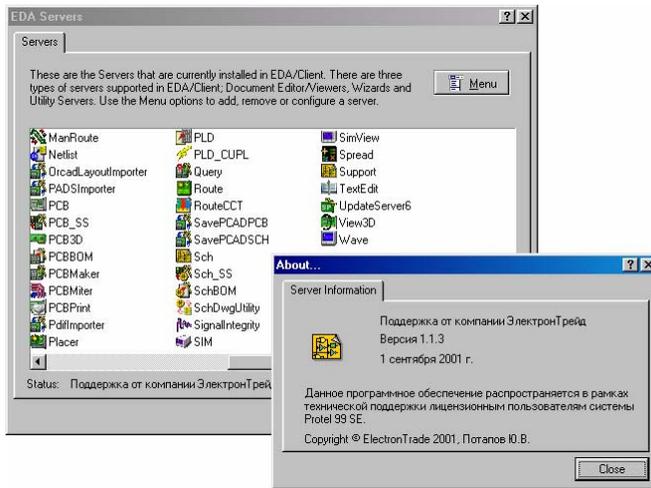


Рис. 2.36. Вид окна About нового сервера.

После сохранения данного файла остается проинсталлировать сервер в среду Design Explorer. В строке Status, расположенной в нижней части диалогового окна **EDA Servers**, подтверждается корректность инсталляции выбранного сервера, а также его текущее состояние. Так сервер не содержит программного кода, лишь управляет ресурсами его не обязательно запускать или останавливать. Удаление сервера из системы производится с помощью команды Remove из меню, появляющегося при нажатии кнопки **Menu** в окне **EDA Servers**. Команда **About** позволяет посмотреть информацию о версии ново сервера (рис. 2.36).

Автоматическая инсталляция сервера

В некоторых случаях необходимо выполнить автоматическую инсталляцию сервера обновления ресурсов в среду проектирования Design Explorer. Так работают модули Service Pack обновления автоматического обновления системы Protel 99 SE.

Для автоматической инсталляции сервера в систему необходимо с помощью внешней программы инсталляции открыть файл **Client99SE.ini**, расположенный в папке Windows, и в его конец добавить две строки (применительно к созданному ранее серверу):

```
[New Servers]
Support=C:\Program Files\Design Explorer 99 SE\Support\Support.ins
```

При первом же запуске системы Protel 99 SE, данный сервер будет найден и автоматически проинсталлирован в систему, а указанная строка перезаписана в нужное место. После этого с данным сервером можно производить все стандартные операции (восстановление ресурсов, деинсталляцию и т.д.).

Текстовый редактор системы Protel 99 SE

В состав системы Protel 99 SE входит мощный текстовый редактор Text Editor, позволяющий создавать и редактировать различные документы в ASCII формате не покидая среды проектирования Design Explorer. Здесь можно производить все базовые операции с текстовыми документами такими, как отчеты, макросы, исходные файлы проектов на базе ПЛИС.

Внутри текстового редактора можно производить привычные операции редактирования: выделение, вырезание, копирование, вставку, поиск и замену текстовых фрагментов. Одной из наиболее мощных функций текстового редактора является возможность цветовой подсветки текста **Syntax Highlighting** в зависимости присутствующих в тексте слов, символов и идентификаторов. Эта функция особенно полезна при работе со сложными структурированными текстовыми документами, например, с текстами макросов или исходными файлами проектов на ПЛИС.

Опираясь на синтаксические особенности различных языков программирования, можно сконфигурировать выделение цветом различных частей документов. Внутри каждого языка могут быть назначены собственные ключевые слова, строки, числа, комментарии и идентификаторы, которым будет присвоен свой цвет.

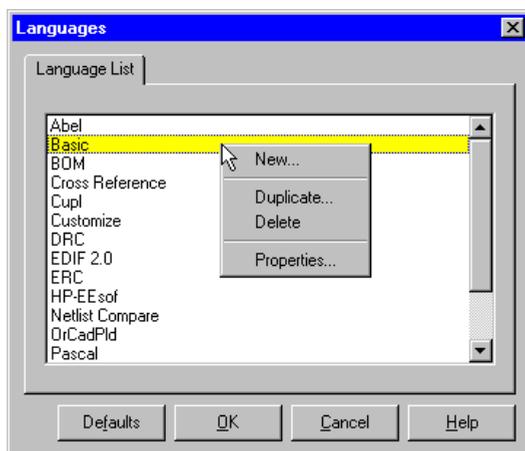


Рис. 2.37. Выбор языка для проверки синтаксиса в текстовом редакторе.

Языки программирования и описания

В текстовый редактор уже введены сведения о многих существующих языках, а также имеется возможность создания нового языка. "Новизна" языка определяется здесь

скорее не наличием непохожих слов, как в обычной разговорной речи, а наличием индивидуальных синтаксических правил, что наиболее часто встречается в языках программирования.

Языки можно создавать, редактировать и удалять в диалоговом окне **Languages**, вызываемом командой меню текстового редактора **Tool » Change Language** или кнопкой **Change Language** на панели инструментов (рис. 2.37). Выберите нужный язык из списка и нажатием правой кнопки мыши вызовите контекстное меню работы с ним.

Редактор распознает языки по расширению текстовых файлов, которое назначается в поле **Associated Files** в окне **Edit Syntax**, вызываемом двойным щелчком левой кнопкой мыши на нужном языке из списка. При наличии нескольких расширений, они все должны быть перечислены в этом поле, причем в качестве разделителя должна быть использована запятая.

Функция выделения цветом

Данная функция имеет две основных области применения: редактирование синтаксиса и выделение цветом различных языковых идентификаторов.

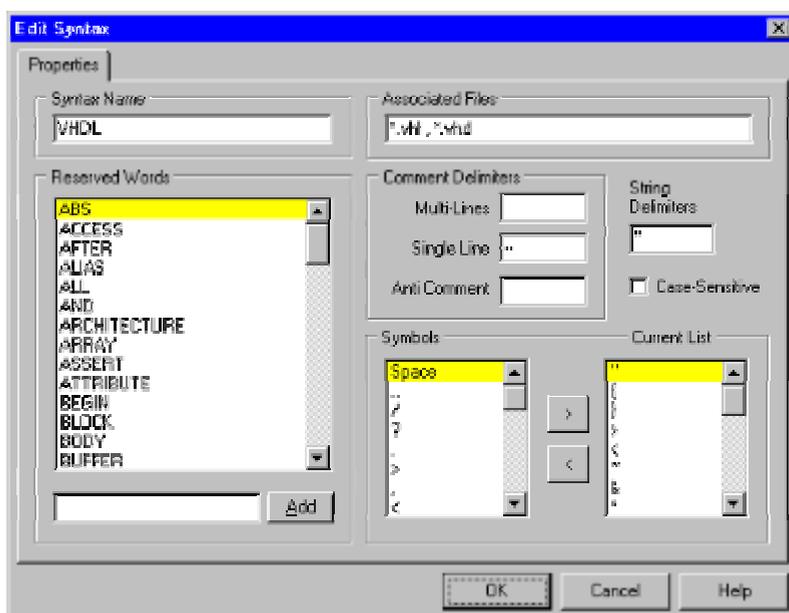


Рис. 2.38. Синтаксис для выбранного языка определяется в диалоговом окне Edit Syntax

Для редактирования синтаксиса необходимо выполнить команду меню **Options » Change Language**, вызывающую диалоговое окно **Languages**, в котором следует вы-

брать язык и нажать кнопку **Edit**. Появится диалоговое окно **Edit Syntax** (рис. 2.38), в котором задаются наборы резервированных слов, принципы ограничения командных строк и комментариев, допустимые наборы символов и расширения файлов, ассоциированные с данным языком. Цвета подсветки задаются в диалоговом окне **Text Editor Properties**, вызываемом командой меню **Tools » Preferences**.

Параметры документов

Для изменения параметров документов необходимо выполнить команду меню **Tools » Options**, после чего появится диалоговое окно **Text Editor Properties**. Здесь на вкладке **Text Colors** можно задать цвета для выделения различных элементов текста.

Работа с макросами

Среда проектирования Design Explorer включает мощный сервер **Macro**, позволяющий записывать, отлаживать и выполнять макросы. Поддерживаемый этим сервером язык называется Client Basic и является подмножеством языка Visual Basic.

Макросы представляют собой мощный механизм автоматизации труда разработчика, а значит, и повышения эффективности проектирования. Сервер макросов поддерживает все доступные среде проектирования Design Explorer процедуры и обеспечивает возможность передачи в них соответствующих наборов параметров. Макрос может быть написан для работы с любым подключенным к системе сервером.

Макросы могут выполнять любые сколь угодно сложные повторяющиеся последовательности процедур, с прохождением всех присутствующих в них диалоговых окон. Сервер макросов также поддерживает технологию OLE, благодаря чему имеется возможность выполнения операций другими совместимыми приложениями системы Windows.

Интерпретатор языка Client Basic позволяет выполнять только что написанный код, минуя операцию компиляции. Аналогично любой другой процедуре макрос может быть запущен с помощью любого ресурса оболочки Design Explorer. Сервер макросов включает все необходимые средства отладки, такие как точки прерывания, пошаговое выполнение, просмотр значений переменных, а также систему поиска и исправления ошибок.

Если в тексте макроса встречается ошибка, выполнение его прерывается, текст открывается текстовым редактором, строка с ошибкой выделяется цветом, а в специальном окне выдается сообщение об ошибке и ее краткое описание.

Пример программирования на языке Client Basic

Для начала создадим наше первое приложение. Все, что оно будет делать — это выводить окно с сообщением "Hello, World!". Несмотря на столь малую функциональность, оно позволит нам опробовать некоторые приемы работы со средой создания макросов.

1. Запустите систему Protel 99 SE. Закройте все загруженные проекты командой **File » Close Design**.
2. Создайте новый проект командой **File » New**. В поле **Database File Name** введите имя нового проекта **Hello.ddb**. Нажмите кнопку **OK**. Система создаст новый проект **Hello.ddb**, изменит главное меню и панель инструментов.
3. Щелчком левой кнопкой мыши на значке папки Документы откройте ее. Создайте новый текстовый документ в этой папке, для чего выполните команду меню **File » New...** или команду **New** из всплывающего меню, вызываемого щелчком правой кнопкой мыши в поле папки. В открывшемся окне **New Document** на закладке **Documents** выберите значок **Text Document** и нажмите кнопку **OK**.
4. Система запросит имя файла текстового документа. Введите **Hello.bas**. В общем случае расширения имен файлов в системе Protel не играют никакой роли, однако расширение **.bas** у текстового документа, позволит системе автоматически ассоциировать этот документ с языком Client Basic.

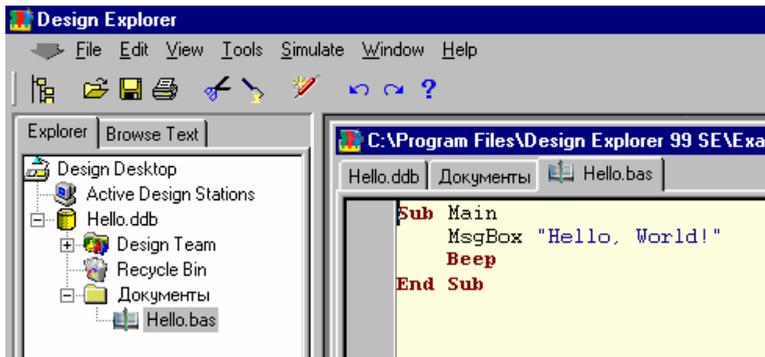


Рис. 2.39. Вид текстового редактора в режиме написания макросов.

5. Сделайте двойной щелчок на имени **Hello.bas**. Система откроет этот документ в режиме редактирования текста, а также соответствующим образом изменит меню команд и панель инструментов.
6. Введите текст макроса, приведенный ниже. Обратите внимание, что система автоматически выделяет ключевые слова языка Client Basic (рис. 2.39).

```
Sub Main
  MsgBox "Hello, World!"
  Beep
End Sub
```

7. Сохраните изменения в файле с помощью команды главного меню **File** » **Save**.
8. Запустите макрос на выполнение командой системного меню **Client** » **Run Script...** В появившемся окне **Select** выберите имя файла макроса **Hello.bas** и нажмите кнопку **OK**.

В результате выполнения макроса на экране монитора будет отображено окно сообщения "Hello, World!" (рис. 2.40).



Рис. 2.40. Сообщение, полученное в результате выполнения макроса.

Указанный метод запуска макроса на выполнение слишком неудобен для частого использования. Альтернативой команде запуска макроса **Run Script...** является привязка макроса к так называемому “пускателю” процесса. Пускателем процесса в системе Protel 99 SE является элемент ресурсов пользовательского интерфейса: кнопка панели управления, горячая клавиша или команда меню. Во всех трех случаях макрос запускается процессом **Macro:RunMacro**, который в свою очередь связывается с пускателем. В качестве примера рассмотрим привязку нашего макроса **Hello.bas** к комбинации горячих клавиш **ALT+1**.

1. При открытом тексте макроса **Hello.bas** выполните команду меню **Client** » **Customize...** и выберите закладку **ShortCut Keys** в диалоговом окне **Customize Resources**.
2. Нажмите на кнопку **Menu** и выполните команду **Edit...** В результате появится окно **ShortCut Tables**. В поле **Name** отображается имя таблицы горячих клавиш. Убедитесь, что оно равно значению **Default**.
3. Снова нажмите кнопку **Menu** и выполните команду **Add**. В результате в таблице горячих клавиш появится заготовка для процесса с заголовком **[None]**.
4. Чтобы настроить процесс и связать с ним горячую клавишу, сделайте двойной щелчок на заголовке процесса. Появится диалоговое окно **ShortCut**, показанное на рисунке 2.41, в котором в строке **Process** следует указать имя процесса **Macro:RunMacro**, выбрав его из выпадающего списка. В строку **Parameters** введите следующие параметры процесса:

```
DesignName=\Examples\Hello.ddb | DocumentName=Документы\Hello.bas |  
Language=ClientBasic
```

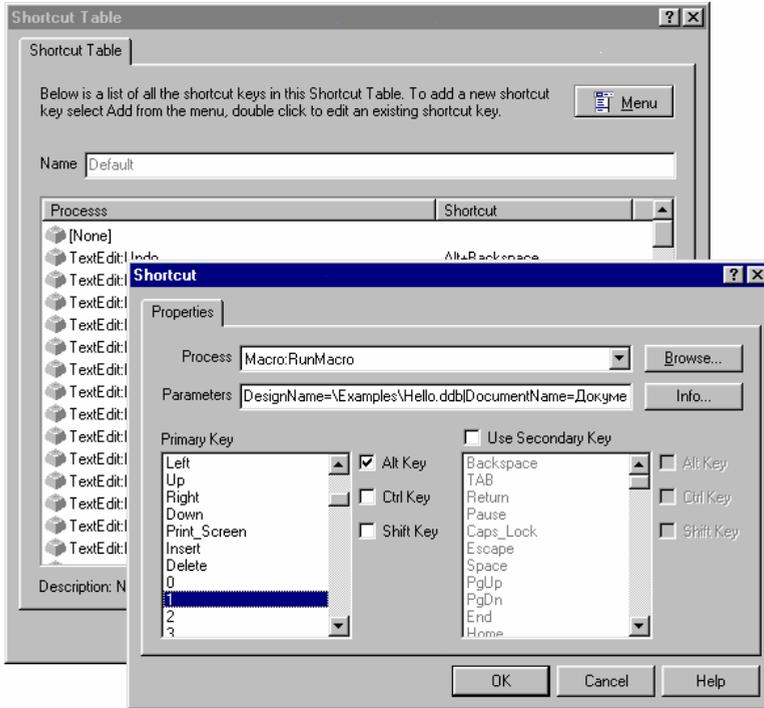


Рис. 2.41. Назначение макросу горячей клавиши.

Напомним, что параметры процесса разделяются символом вертикальной прямой черты "|". В нашем случае первый параметр означает имя проекта, второй — имя документа в проекте с указанием полного пути к нему (в нашем случае — папка **Документы**), третий — используемый язык программирования.

5. Установите флажок **Alt Key** и выберите из списка **Primary Key** клавишу **1**.
6. Закройте окно **ShortCut**, нажав кнопку **OK**, и окна **ShortCut Table** и **Customize Resources**, нажав соответствующие кнопки **Close**.

Теперь при нажатии комбинации клавиш **ALT+1** при активном текстовом файле на экран монитора будет выводиться сообщение "Hello World!" — результат выполнения макроса **Hello.bas**. Однако, данная комбинация горячих клавиш будет работать только в текстовом редакторе, так как мы производили настройку именно его ресурсов.

Пользу от практического применения макросов совместно с пользовательскими настройками ресурсов можно проиллюстрировать примером переключения сеток

SnapOn и **Visible** в редакторе принципиальных схем. Для изменения настроек этих сеток обычным способом необходимо выполнить ряд последовательных операций:

1. При активной принципиальной схеме командой меню **Design** » **Options** открыть диалоговое окно **Document**.
2. Изменить настройки сеток в поле **Grids** вкладки **Sheet Options** (рис. 2.42).
3. Закрыть окно нажатием кнопки **OK**.

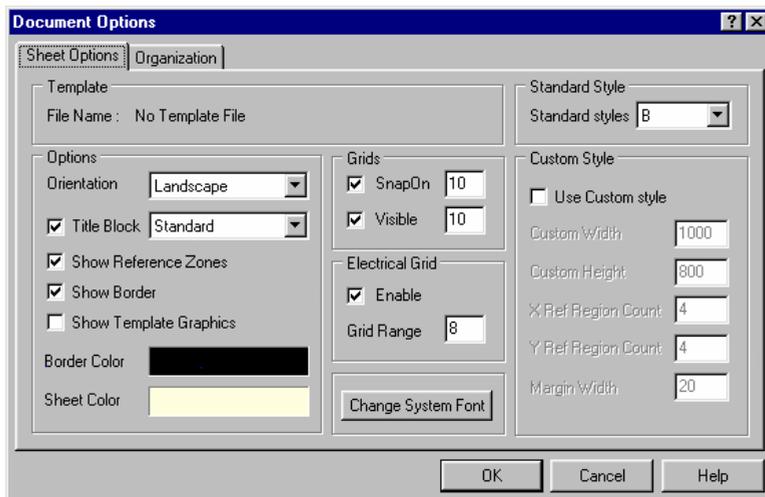


Рисунок. 2.42. Традиционный способ настройки сеток редактора схем.

Столь сложная процедура настройки сеток может вызвать обоснованное недовольство пользователей, кто ранее работал в редакторе принципиальных схем пакета P-CAD 2001 и переключал шаг сетки одним щелчком, выбирая нужное значение из соответствующего выпадающего списка. Ведь в процессе разработки сложной схемы эту операцию приходится выполнять сотни раз! Однако, не все так плохо, как кажется на первый взгляд.

Создадим макрос с приведенным ниже текстом и привяжем его выполнение к новой кнопке на главной панели инструментов редактора схем.

```
Sub Main
  ResetParameters
  AddIntegerParameter "SnapGridSize", 2
  AddIntegerParameter "VisibleGridSize", 10
  RunProcess "Sch:DocumentPreferences"
  beep
End sub
```

Данный макрос автоматически задает шаг сетки **SnapOn** равный двум дискретам, шаг сетки **Visible** равный десяти дискретам. Об успешном завершении выполнения макроса сигнализирует оператор **Beep**, который здесь не является обязательным. В состав сервера поддержки, поставляемого вместе с продуктом Protel 99 SE входят три макроса, различным образом изменяющие настройки сеток всего за один щелчок по соответствующей кнопке (рис. 2.43).

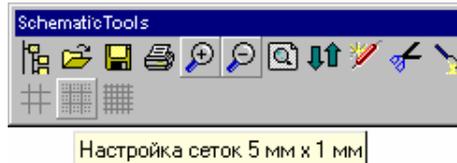


Рисунок. 2.43. Кнопки настройки сеток с помощью макросов.

После рассмотренных примеров у пользователя наверняка появится желание написать и отладить свой собственный макрос. А чем сложнее макрос, тем больше вероятность наличия в нем синтаксических ошибок, делающих выполнение макроса невозможным, а иногда и зависающих систему. Поэтому отдельно следует остановиться на встроенных в текстовый редактор средствах отладки.

Как показано выше, все макросы в системе Protel создаются с помощью текстового редактора. Функции отладки макросов добавляются в него с помощью сервера **Macro**, входящего в комплект стандартной поставки и автоматически устанавливаемого в систему. Команды управления отладчиком будут находиться в разделе меню **Macro**. Однако при обычной инсталляции в среду проектирования Design Explorer не добавляется специальная панель инструментов, дающая пользователю более удобный доступ к средствам отладчика.

Чтобы вручную подключить эту панель к системе необходимо выполнить следующее:

1. Выполнить команду **Client » Customize...** и в появившемся окне **Customize Resources** выбрать вкладку **Toolbars**.
2. Нажать кнопку **Menu** и выбрать команду **Import From Other Document**.
3. В открывшемся окне **Resource Allocation For Document Type** (рис. 2.44) в левом списке **Resources** следует выбрать пункт **MacroDebugTools** и нажать кнопку **Add**. После этого панель управления **MacroDebugTools** окажется в списке справа, где отображаются доступные в настоящий момент панели инструментов.
4. Закрывать все окна нажатием кнопок **Close**.

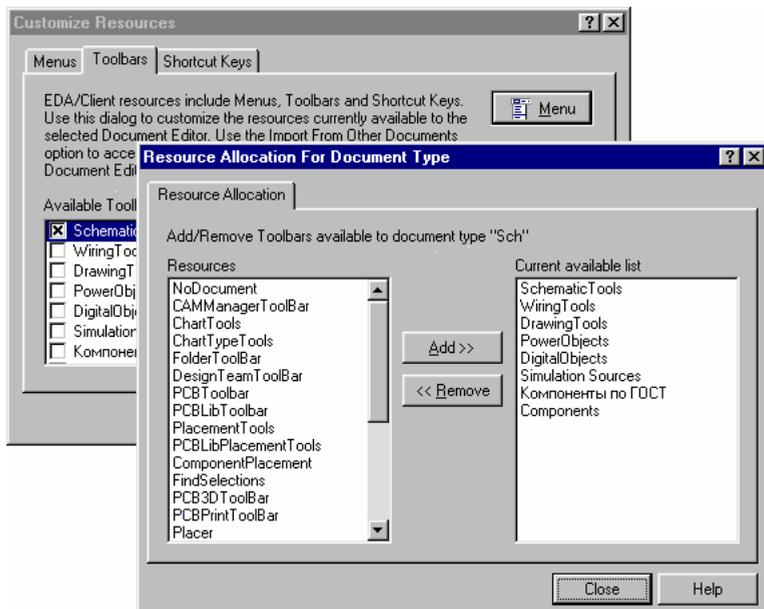


Рис. 2.44. Подключение панели инструментов отладки макросов.

После этого в разделе меню **View** текстового редактора появится новая команда **Toolbars » Macro Debug Tools**, включающая панель инструментов, изображенную на рисунке 2.45. Имеющиеся здесь кнопки дублируют команды меню из раздела **Macro**:



Рис. 2.45. Вид панели инструментов отладки макросов.

- View Breakpoint Window — включить/выключить окно просмотра точек прерывания.
- View Watch Window — включить/выключить окно просмотра значений переменных.

- Run Client Basic Dialog Editor — запустить редактор диалоговых окон.
- Reset — сброс выполнения активного макроса.
- Go — запуск выполнения активного макроса.
- Pause — приостановление выполнения активного макроса.
- Step Into — войти в функцию или подпрограмму активного макроса.
- Step Over — пошаговое выполнение активного макроса.
- Add Watch — добавить переменную к списку наблюдаемых.
- Remove Watch — удалить переменную из списка наблюдаемых.
- Remove All Watch — удалить все переменные из списка наблюдаемых.
- Add Breakpoint — добавить точку прерывания.
- Remove Breakpoint — удалить точку прерывания.
- Remove All Breakpoint — удалить все точки прерывания.
- Toggle Animate Mode — включить/выключить анимационный режим отображения хода выполнения макроса.

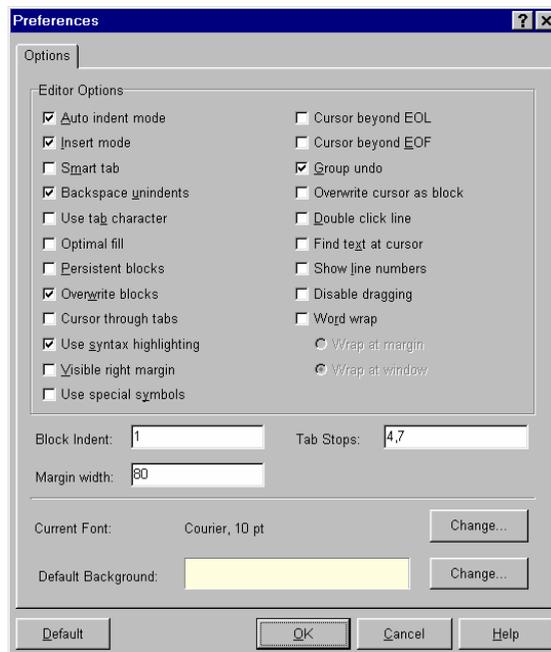


Рис. 2.46. Настройка текстового редактора.

С помощью описанных функций можно отлаживать работу сколь угодно сложного макроса, а также создавать диалоговые окна с самым разнообразным набором органов управления.

Поведение текстового редактора настраивается с помощью установок диалогового окна **Preferences**, вызываемого командой меню **Tools » Options** (рис. 2.46). Здесь имеются следующие опции:

- **Auto indent mode** — при установленном флажке каждая новая строка будет иметь такой же отступ от левого края, как и предыдущая; при сброшенном флажке новая строка будет начинаться от левого края;
- **Insert mode** — при установленном флажке редактор работает в режиме вставки текста; при сброшенном — в режиме замены;
- **Smart Tab** — при установленном флажке и сброшенном флажке **Use tab character** курсор при нажатии клавиши **Tab** будет перемещаться на позицию начала следующего слова предыдущей строки; при сброшенном — на расстояние табуляции;
- **Backspace unindents** — при установленном флажке и установленном флажке **Smart Tab**, при нажатии клавиши **Backspace**, отступ строки от левого края будет смещаться в соответствии с позицией начала слов предыдущей строки; при сброшенном — на один символ влево;
- **Use tab character** — при сброшенном флажке символы табуляции будут заменяться пробелами;
- **Optimal fill** — используется совместно с **Use tab character**. При формировании отступа (используется **Auto indent mode**) — используется минимальное количество символов табуляции и пробелов;
- **Persistent blocks** — при установленном флажке блок остается выделенным до начала операции следующего выделения или щелчка мыши; при сброшенном флажке — любая команда перемещения курсора сбрасывает выделение блока;
- **Overwrite blocks** — при установленном флажке вводимый текст заменяет выделенный;
- **Cursor through tabs** — при установленном флажке курсор перемещается на расстояние символов табуляции; при сброшенном флажке — на расстояние пробела;
- **Use syntax highlighting** — при установленном флажке ключевые слова языка будут выделяться цветом;
- **Visible right margin** — при установленном флажке будет отображаться линия правой границы листа;
- **Use special symbols** — при установленном флажке будет отображаться специальные символы, такие как конец строки EOL и конец документа EOF.
- **Cursor beyond EOL** — при установленном флажке можно перемещать курсор за пределы конца строки;

- Cursor beyond EOF — при установленном флажке можно перемещать курсор за пределы конца документа;
- Group undo — при установленном флажке все однотипные операции будут отменяться за один шаг;
- Overwrite cursor as block — при установленном флажке в режиме замены текста курсор будет отображаться как мерцающий блок; при сброшенном флажке — как мерцающая линия;
- Double click line — при установленном флажке двойной щелчок левой кнопкой мыши приведет к выделению текущей строки; при сброшенном флажке — текущего слова;
- Find text at cursor — если флажок установлен, то при вызове команды **Edit » Find** в поле **Text to find** диалогового окна **Find Text** поместится текущее слово;
- Show line numbers — при установленном флажке будут отображаться номера строк;
- Disable dragging — при установленном флажке текст не может перетаскиваться из одного места документа в другое;
- Word wrap — при установленном флажке текст будет переноситься на другую строку при достижении границы окна;
- Block Indent — задает число символов на которое будет смещаться блок текста при операциях Indent/Unindent;
- Margin width — определяет правую границу страницы в символах текста;
- Tab stops — задает интервалы табуляции, причем разность между двумя последними значениями определяет интервалы для последующих табуляций;

Следует отметить одно неудобство системы Protel 99 SE, сильно осложняющее работу с макросами. Так как в системе отсутствует макрорекордер, позволяющий записывать и в последствии автоматически повторять все выполненные пользователем действия (аналогично программам MS Word и Excel), от пользователя требуется безукоризненное знание всех тонкостей языка Client Basic.

Глава 3

Разработка принципиальных схем

Основные возможности редактора принципиальных схем

В этой главе описываются основные возможности редактора электрических принципиальных схем системы Protel 99 SE (рис. 3.1). Здесь приведена информация, необходимая при работе с системой, описаны основные функции программы по разработке схем, подготовке к проверке правил электрических соединений, генерации списка соединений и выводу конструкторской документации на печать. Также приведены рекомендации по оформлению схем для наибольшего соответствия российским стандартам.

Редактор электрических принципиальных схем

Редактор схем позволяет создавать, редактировать, проверять и распечатывать листы схем, составляющих конструкторский проект. В редакторе доступны все инструменты и утилиты, необходимые для подготовки к проверке правил электрических соединений, генерации отчетов и создания презентабельного качества чертежа схемы.

Редактор библиотек элементов принципиальных схем

Редактор библиотек элементов принципиальных схем необходим для создания, редактирования и управления библиотеками компонентов. В редакторе схем термин "элемент" ("part") обозначает условное графическое обозначение (УГО) того или иного элемента принципиальной схемы или его секции (например, один элемент 2И/НЕ в микросхеме 7400). Редактор библиотек элементов включает в себя основную массу функций редактора принципиальных схем, а также специальные инструменты и функции для создания и описания секций компонентов, а также решения задач управления библиотеками.

Возможности проектирования

Редактор электрических принципиальных схем представляет собой комплексный инструмент ввода и разработки принципиальных схем различных устройств. В комбинации с программой моделирования Protel Circuit Simulator, модулем разработки устройств на ПЛИС (PLD) и этот модуль становится частью мощной, полностью ав-

томатизированной, интегрированной системы сквозного проектирования радиоэлектронных устройств.

Редактор схем позволяет разрабатывать многострочные и иерархические проекты, размер которых ограничен только размером оперативной памяти и емкостью жесткого диска компьютера. Листы схем могут иметь стандартные, например, А0 — А4, или заданные пользователем размеры. Также существует возможность создавать пользовательские рамки, основные надписи и сохранять созданные самостоятельно шаблоны для их повторного использования.

Поддержка библиотек элементов

Стандартная библиотека компонентов редактора принципиальных схем содержит свыше 65.000 компонентов. Специальное подразделение компании Altium постоянно обновляет существующие библиотеки элементов, которые можно найти на интернет-сайте фирмы, находящемуся по адресу www.protel.com.

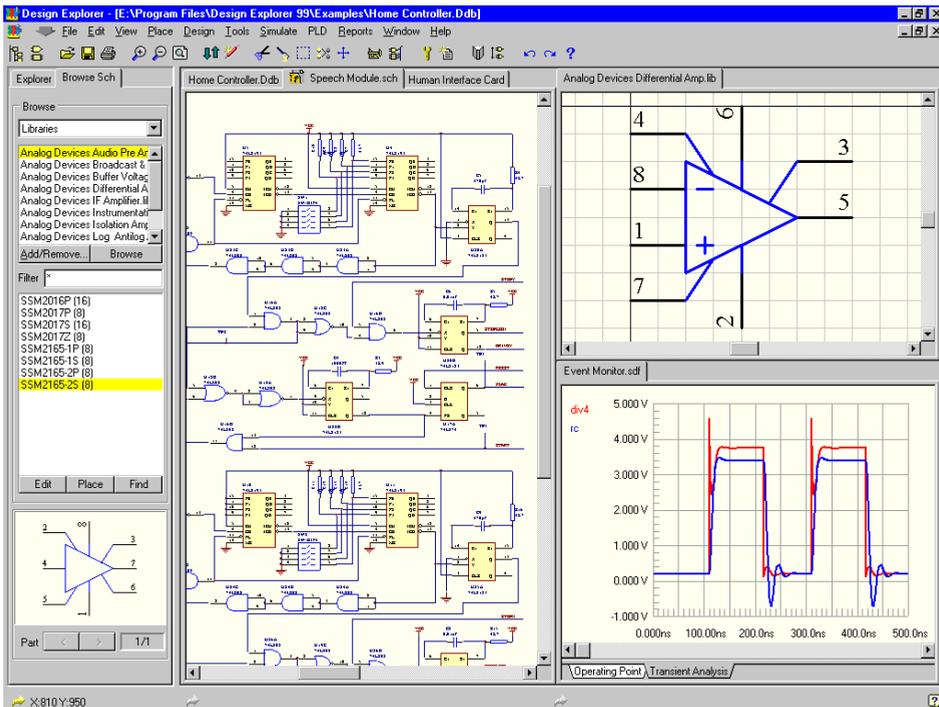


Рис. 3.1 Внешний вид редактора схем системы Protel 99 SE.

Поддержка иерархических и многолистовых проектов

Редактор принципиальных схем поддерживает проекты, состоящие из одного или нескольких листов, а также имеющие иерархическую структуру с многократным вложением. Управление таких проектов осуществляется с помощью специального инструмента — окна просмотра дерева проекта Design Explorer. Он в удобной и наглядной форме показывает все листы проекта с учетом связей между ними. Переход от одного листа проекта к другому здесь осуществляется одним щелчком кнопки мыши.

Интеллектуальные функции задания связей

Специальные автоматизированные функции ускоряют процесс соединения электрических элементов на листе принципиальной схемы. Электрическая сетка обеспечивает соединение всех электрических элементов: портов, входов листа, шин, входов шин, идентификаторов цепей, проводников и элементов схемы. Когда эта функция активна, курсор прыгает к ближайшей электрической "горячей" точке, находящейся в диапазоне действия сетки, и затем изменяет свою форму, показывая таким образом точку соединения. Для завершения операции соединения нужно всего лишь щелкнуть кнопкой мыши (или отпустить левую кнопку мыши, если производилось перемещение объекта).

При перемещении электрических объектов на принципиальных схемах соединения между ними сохраняются. Система автоматически добавляет или удаляет сегменты проводников для поддержания ортогональности разводки даже при достаточно сложных перемещениях.

Гибкая система выбора объектов

Для удобства работы имеется гибкая система выбора групп элементов не только как нарисованных объектов, но и как физически связанных элементов. В уже созданное выделение могут быть добавлены отдельные элементы (или удалены из него). С группами элементов можно работать, используя стандартные команды правки: **Cut** (вырезать), **Copy** (копировать), **Paste** (вставить) или **Clear** (удалить). Группу можно вращать, перемещать, а также копировать в буфер обмена для последующей вставки в другие приложения. Еще одной полезной функцией является возможность копирования в буфер обмена шаблона, содержащего весь лист, включая рамку и основную надпись.

Мощные возможности редактирования схем

Любой объект, входящий в проект, имеет атрибуты, которые можно отредактировать через специальное диалоговое окно. Такое окно появляется при двойном щелчке мыши на интересующем объекте. Можно изменить атрибуты не только одного объекта, но и всех объектов подобного типа на листе или во всем проекте. Например, при редактировании проводников можно изменить такие их атрибуты, как цвет или размер. Изменять проводники можно как на текущем, так и на всех других открытых листах проекта. Аналогичные средства редактирования существуют для элементов схем и других графических объектов.

Для редактирования текстовой информации реализованы удобные функции **Find** (найти) и **Replace** (заменить), область действия которых распространяется на все листы проекта.

Система библиотек элементов

Редактор принципиальных схем включает комплексные инструменты для управления библиотеками компонентов. Не выходя из редактора листов, можно работать с любым числом библиотек.

Элементы могут быть выбраны и вставлены в схему непосредственно из редактора библиотек элементов. Вместе с системой поставляется большой библиотек стандартных компонентов различных производителей. При сетевой установке имеется возможность одновременного доступа нескольких пользователей к одной и той же библиотеке. Изменение какого-либо элемента в библиотеке отражается в проекте сразу после выполнения команды обновления содержимого экрана.

Компоненты включают 8 неизменяемых (библиотечных) текстовых полей и 16 редактируемых текстовых полей, размером до 255 символов каждое. Для удобства работы в редакторе библиотек возможность предопределить имена этих полей для различных типов компонентов.

Специальные строки

Для размещения на листах принципиальных схем дополнительной информации (даты, название листа, имени файла, счетчика компонентов и др.) служат специальные строки. Например, размещение на листе строки **".DATE"** обеспечит вывод на печать в этом месте схемы текущей системной даты. Специальные строки могут быть внедрены и в шаблоны листов принципиальных схем.

Поддержка шрифтов

Система Protel 99 SE полностью поддерживает шрифты TrueType. Системные шрифты могут быть определены для выводов компонентов, имен портов и объектов, а также для текстовых полей на листе принципиальной схемы. Для остальных объектов, включающих в себя текстовые поля шрифт задается по умолчанию.

Функция размещения массивов элементов

Эта функция позволяет размещать наборы элементов с упорядоченным шагом. Количество копий элементов, шаг по различным координатным осям и способ изменения имен дубликатов объектов задаются непосредственно пользователем.

Функция выравнивания элементов

Объекты на принципиальных схемах могут быть выровнены по левой, правой, верхней или нижней границам, равномерно распределены по горизонтали или по вертикали, перемещены в узлы заданной сетки.

Инструменты верификации проекта

Функция проверки правильности электрических соединений (Electrical Rule Check) позволяет быстро проверить даже сложные многолистовые проекты. Проверка выполняется в соответствии с определенными пользователем физическими и логическими параметрами. Эта функция поддерживает расстановку флагов и генерацию отчетов по большому числу предупреждений о физических и логических ошибках, таких как несоединенные метки цепей, незапитанные элементы и неиспользуемые входы элементов.

Вывод схем на печать

С помощью одного общего для всех функций печати пункта меню Print можно управлять печатью на матричных и лазерных, черно-белых и цветных принтерах, выводом схем на плоттер или в формате PostScript, т. е. на любое устройство, которое поддерживается операционной системой Windows. Напечатанные схемы при этом имеют отличное качество и читаемость.

Основы разработки принципиальных схем

Создание принципиальных схем представляет собой процесс описания проекта на схемотехническом уровне для последующей разработки его с применением компьютера. Определение проекта на схемном уровне позволяет работать с логической моделью проекта, тогда как конечным продуктом разработки является полноценная физическая модель устройства.

Представление электрической схемы в компьютере

На первый взгляд разработка принципиальной схемы похожа на процесс создания обычного чертежа, где на схему наносятся графические объекты, соответствующие различным элементам электрических цепей, являющихся частями проекта. Использование компьютера здесь дает множество преимуществ за счет автоматизации процесса создания чертежа и значительного облегчения редактирования проекта.

Однако, именно на этапе ввода схемы обеспечивается главное преимущество — создание интегральной связи между концепцией схемы и ее физическим воплощением. Заложенная "логика" схемы обеспечивает возможность интеграции в пределах проекта механизмов моделирования и физической компоновки элементов. Процесс схемного проектирования становится отправной точкой для использования некоторых других технологий — от построения интегральных схем до программирования ПЛИС (FPGA и PLD), моделирования схем и проектирования печатных плат.

Редактор принципиальных схем объединяет в себе множество различных функций управления данными. Например, каждый лист принципиальной схемы является независимым документом. Автоматизированная система соединяет эти листы в иерархическую структуру, которая и составляет проект. Теперь разработчик имеет возможность редактировать и выполнять проверку сразу всех листов, а также создать список материалов для проекта целиком с помощью всего одной команды.

Модели элементов принципиальных схем

Элементы принципиальных схем объединены в наборы библиотек, построенных на справочниках производителей электронных компонентов.

Компонент библиотеки описывает элемент или его отдельную секцию, и это описание является представлением данного компонента на листе принципиальной схемы.

Так как компоненты могут иметь несколько однородных или неоднородных секций (например, несколько вентилях в логической микросхеме), дополнительное удобство происходит из возможности работы с каждой секции по отдельности.

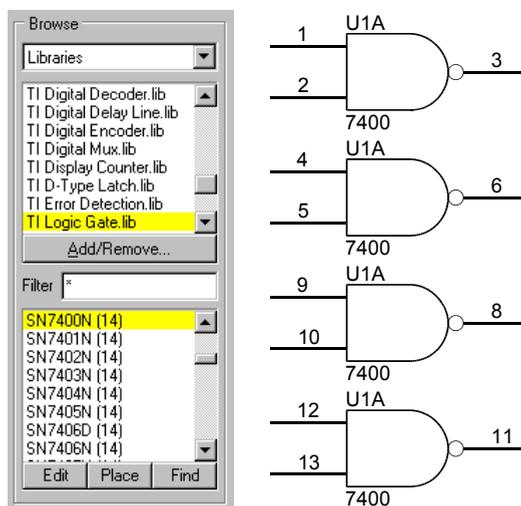


Рис. 3.2. Компонент может состоять из нескольких, иногда неодинаковых секций.

Библиотеки моделей компонентов

Если условное графическое изображение какого-либо компонента наносится на схему, то вся доступная информация об этом компоненте всегда сохраняется в соответствующей библиотеке. Создание и редактирование элемента принципиальной схемы всегда выполняется на уровне библиотеки, а не на схеме. Такой подход обеспечивает целостность библиотеки и позволяет при внесении в нее изменений глобально обновлять элементы в уже разработанных схемах.

Одиночный библиотечный компонент может включать описания нескольких графических элементов, как показано на рисунке 3.2 на примере простейшей ТТЛ микросхемы. Отдельные элементы — секции компонента могут быть размещены на листах схем независимо от других.

Если какой-либо элемент размещаются на схеме, то его графическое представление помещается в специальный кэш. Этот кэш работает в памяти как резидентная биб-

лиотека. При сохранении листа схемы все используемые элементы копируются из кэша и присоединяются к файлу принципиальной схемы, в виде не редактируемой локальной минибibliotheki, которая позволяет формировать единую библиотеку для данного проекта. При последующем открытии листа описания компонентов считываются из этой библиотеки обратно в кэш. Образы компонентов, сохраненные в локальной библиотеке (и в кэше), могут быть обновлены из исходных библиотек источника в любой момент, пока данный лист загружен в редактор схем. При этом выдается отчет о всех обновленных элементах в кэше.

Связанность

Еще одной важной функцией редактора принципиальных схем является использование связанности. Связанность — это способность программного обеспечения распознавать физические соединения между объектами на листе и ставить им в соответствие логические связи между различными листами в многолистовых проектах. Связанность также используется для скрепления определенных объектов друг с другом. Например, при перемещении электрически связанных элементов (шин, проводников, портов и т.д.) связь между ними не нарушается. Очень важно, что функция связанности позволяет при проектировании схемы генерировать список соединений, который затем передается в редактор печатных плат для выполнения проверки правил электрических соединений (ERC).

Основные электрические объекты, используемые в схеме, включают: проводники — специальные линии, по которым проходят сигналы или подводится напряжение питания; шины — графические представления групп проводников; входы шин — графические представления точки входа проводника в шину; элементы — условное графическое обозначение компонентов схем и их выводов.

Существует еще два особых класса электрических объектов. К первому относятся директивы (**Directive**), применяемые для обозначения незадействованных выводов, после чего механизм проверки правил электрических соединений (ERC) такие выводы пропускает, и топологические атрибуты некоторых цепей, используемые на этапе разработки печатной платы. Ко второму классу относятся идентификаторы цепей, используемые для обозначения электрических соединений, которые физически не связаны между собой, например, соединения между разными листами схемы в многолистовом проекте.

Идентификаторы цепей

Как сказано выше, идентификаторы цепей могут создавать соединения, которые физически не связаны проводниками. Такими объектами являются: метки цепи (**net labels**), определяющие общие цепи на листе (или на нескольких листах в многолистовом проекте); порты (**ports**), описывающие соединения цепей между двумя листами; входы листов (**sheet entries**), определяющие соединения в подсхемах; порты питания (**power ports**), обозначающие общие цепи питания (или любые другие определенные пользователем цепи).

К идентификаторам цепей можно причислить и скрытые выводы элементов, так как функционирование таких выводов аналогично портам питания. Скрытые выводы с одинаковыми именами считаются соединенными между собой, а в случае наличия

одноименной цепи, и с этой цепью. Если вывод перестает быть скрытым, то связь теряется и при необходимости ее нужно восстанавливать вручную.

Использование связности

Методы

Связность происходит из расположения на листе определенных электрических объектов и связей в виде проводников и шин. Однако, не все электрические объекты используют свое расположение для определения поведения связей. Некоторые из них используют для этого свое геометрическое положение, а некоторые — логические связи.

Физическая связанность

Физическая связь происходит из физического контакта "горячих" точек каких-либо двух электрических объектов. В простейшем случае, провод касающийся вывода компонента считается физически соединенным с этим выводом, а редактор схем извлекает из этого информацию о логическом соединении.

В общем случае, если "горячие" точки двух любых объектов соприкасаются, они считаются соединенными. Тем не менее, существует несколько специальных правил, относящихся к определенным классам соединений.

Логическая связанность

Логическая связанность следует из наличия идентификаторов цепи (меток цепей, портов, входов листа, выводов питания и скрытых выводов) на листе. Логическая связанность не требует особого геометрического положения объектов или их физического контакта, она зависит от согласования названий цепей, присоединенных к этим объектам на текущем или других листах проекта. Более подробно определение логических связей описано в разделе *Управление многолистовыми и иерархическими проектами* и подразделе *Создание списков соединений*.

Основные правила определения связанности

Электрические объекты считаются соединенными в случае, если когда их электрические "горячие" точки соприкасаются. Специфические способы задания связей описаны ниже. При включенной электрической сетке (**Design » Options**) курсор автоматически прыгнет в ближайшую "горячую" точку и примет вид круга. Если активная "горячая" точка отображается на листе, то при размещении или перемещении электрических объектов щелчок левой кнопкой мыши (или отпускание левой кнопки при перетаскивании) система автоматически произведет соединение с ней.

Специфические правила определения связанности

Связывание проводников

Проводники, концы которых касаются друг друга под любым углом или накладываются друг на друга считаются соединенными. Наложённые проводники с концами,

расположенными в другом месте листа не считаются соединенными. Проводники, пересекающиеся перпендикулярно или соединенные в виде буквы "Т" не считаются соединенными, если только в точку их пересечения не помещена узловая точка.

Связывание проводника с шиной

Шиной называется графическое представление группы проводников. Не смотря на то, что при связывании у шин подсвечиваются "горячие" точки, электрическое соединение в них не происходит. Графически проводники соединяются с шинами с помощью специальных символов шинных входов. Для определения логической связанности здесь необходимо применять метки цепей на обеих сторонах шинного соединения.

Связывание проводника с выводом элемента

Выводы, касающиеся концов проводников под любым углом, считаются соединенными. Выводы, перпендикулярно пересекающие провода, должны соединяться посредством размещения узловой точки в нужном месте. Для автоматического проставления узловых точек в месте пересечения выводов и проводников необходимо включить соответствующую опцию в меню **Tools » Preferences**.

Связывание проводника с портом

Провод, касающийся конца порта, считается присоединенным к нему.

Связывание проводника с входом листа

Провод, касающийся конца входа листа, считается присоединенным к нему.

Связывание шины с объектом

Шина представляет собой всего лишь графическое обозначение группы проводников, которое не имеет каких-либо специфических физических свойств, требующих описания в списке соединений. Логическая связанность здесь определяется метками цепей (смотри пункт *Связывание метки цепи с шиной*). Заметим, что в случае, когда шина подсоединяется к порту, то ее линия должна завершаться на конце порта.

Связывание метки цепи с проводником

Метки цепи связывают проводник с какой-либо цепью. Для получения такой связи необходимо разместить метку цепи вертикально или горизонтально в том же узле сетки, где располагается проводник. Метки могут быть размещены как на горизонтальных или вертикальных линиях, так и на их пересечениях.

Связывание метки цепи с шиной

Шина представляет собой всего лишь графическое обозначение группы проводников, которое не имеет каких-либо специфических физических свойств, требующих описания в списке соединений. Логическая связанность здесь может быть определена только посредством размещения на ней метки цепи. Как правило, эта метка включает все сигналы шины, например, **HA[0...19]**, что означает цепи **HA0, HA1, HA2** и т. д. до **HA19**. При включении опции **Edit » Select » Net** шины и их входы не подсвечиваются.

Связывание вывода с объектом

Выводы могут быть непосредственно присоединены к другим выводам, проводам, меткам цепи, входам листов и портам. Скрытые выводы могут непосредственно соединяться с соответствующими цепями в редакторе библиотек элементов схем (Schematic Library Editor). Нескрытые выводы могут также непосредственно соединяться выводами на других листах.

Связывание вывода с выводом

Выводы считаются соединенными, если они находятся во взаимном контакте по любому углу.

Объекты без проверки правил электрических соединений (ERC)

Объекты без проверки правил электрических соединений (ERC) считаются соединенными с выводами или проводами, если они находятся в контакте с ними.

Верификация проекта

Верификация проекта — это основная операция, предназначенная для проверки правильности выполнения физических (или электрических) и логических соединений в разработанной схеме. В редакторе принципиальных схем имеется возможность производить проверку правильности электрических соединений, в процессе генерации отчетов при запуске модуля проверки Electrical Rule Check (ERC).

Отчет о проверке правильности электрических соединений

Отчет о проверке правильности электрических соединений (ERC) представляет собой список предупреждений об ошибках в электрических (а некоторых случаях и в логических) соединениях в текущем активном проекте. Сюда входят, например, предупреждения о наличии несоединенных входов элементов или замыканий между разными цепями. Для каждого проекта могут быть определены специфические правила проверки. Для отключения генерации предупреждений об ошибках в отдельные места схемы можно включать специальные объекты без проверки правил электрических соединений (**Suppress ERC**).

Тип сообщения "ошибка" или "предупреждение" может быть определен пользователем в графической матрице состояний выводов, портов и входов листа.

На проверку правильности электрических соединений может повлиять то, каким образом в схеме используются идентификаторы цепей. Например, глобальные метки цепи (включенная опция **Net Labels and Ports Global**) объединяют все цепи с одинаковой меткой на всех листах проекта, тогда как локальные метки цепи (включенная опция **Only Ports Global**) объединяют электрические объекты только в пределах одного листа. Имеется возможность задания "области видимости" идентификаторов цепей для системы проверки правильности электрических соединений, аналогично тому, как это делается при генерации списков цепей.

При включении опции **Add Error Markers** в лист вводятся специальные символы, показывающие места упомянутые в отчете о проверке до тех пор пока ошибки не будут исправлены.

Запуск проверки правильности электрических соединений (ERC) — неотъемлемая часть этапа разработки принципиальной схемы проекта. Тщательная проверка и исправление всех ошибок обязательно должно предшествовать передаче информации о схеме в редактор печатных плат.

Проверка листов и проектов

Некоторые функции проверки доступны непосредственно на текущем листе в редакторе принципиальных схем. Для проверки связанности объектов с той или иной цепью может быть использована операция выделения **Edit » Select » Net** и **Edit » Select » Connection** (выводы компонентов при этом не выделяются). Команда **Reports » Selected Pins** позволяет просмотреть список выводов, связанных с выделенной цепью.

Связь между редактором схем и редактором печатных плат

Синхронизация схемы и печатной платы

Система Protel 99 SE имеет мощный механизм синхронизации различных частей проекта, который позволяет автоматически согласовывать данные на принципиальной схеме с данными на печатной плате. Раньше изменения в схемах переносились на печатные платы с использованием так называемого метода прямого аннотирования, а информация об изменениях в печатных платах возвращалась на этап схемного проектирования посредством метода обратного аннотирования. Теперь обе этих функции выполняет синхронизатор.

В процессе работы синхронизатор сначала проверяет наличие компонентов и связей между ними как на листах принципиальных схем, так и на чертежах печатных плат, а затем модифицирует их так, чтобы они соответствовали друг другу. При запуске синхронизатора надо выбрать направление синхронизации включением одной из двух опций **Update PCB from Schematic** (обновить плату исходя из изменений в схеме) или **Update Schematic from PCB** (обновить схему исходя из изменений в плате).

Включение в принципиальные схемы информации для редактора печатных плат

Специальная директива редактора топологий PCB вам определить процесс разводки топологии, приоритеты, ширину проводников и размеры переходных отверстий непосредственно для связей на принципиальной схеме. Эта информация автоматически передается в редактор печатных плат, если синхронизатор настроен на генерацию правил проектирования **Design Rules** из топологических директив **Layout Directives** (для этого надо задать **Design » Synchronizer Options**).

Горячая связь между редакторами схем и печатных плат

Во время работы над принципиальной схемой, если открыт чертеж печатной платы, у пользователя имеется возможность отслеживать переходы между соответствующими друг другу объектами в обеих частях проекта посредством двунаправленного механизма горячей связи (**Cross Probing**).

Например, после нажатия на кнопку включения горячей связи (**Cross Probe**) вы можете выбрать элемент или группу элементов на принципиальной схеме, а редактор печатных плат подсветит соответствующие компоненты на плате. Аналогичным образом поддерживается еще два вида перекрестного тестирования: между выводом на схеме и контактной площадкой на топологии, а также между меткой связью на схеме и проводником.

Настройки редактора принципиальных схем

Параметры редактора

Параметры редактора принципиальных схем задаются в диалоговом окне, появляющемся после выполнения команды меню **Tools » Preferences**. Данное диалоговое окно имеет три вкладки.

Вкладка Schematic

Pin Options

Параметры поля **Pin Options** определяют местоположение названий и номеров выводов элементов схем. **Pin Name Margin** определяет расстояние между названием и выводом, а **Pin Number Margin** — расстояние между номером и выводом элемента в сотых долях дюйма.

Auto-Junction

Данный флаг включает или выключает режим соединения связей при наложении конца связи на уже существующую связь. Во включенном режиме система автоматически поставит точку, обозначающую соединение цепей.

Drag Orthogonal

Включает такой режим перетаскивания компонентов, при котором все связи остаются ортогональными, то есть углы сохраняются кратными 90°. При выключении этого флажка связи могут располагаться под произвольным углом.

Multipart Suffix

Каждая секция составного компонента может обозначаться с помощью специального суффикса, например, **U1:1**, **U1:2** и т. д. или **U1A**, **U1B** и т. д. Обратите внимание, что это глобальная установка, которая будет распространяться на все открытые в настоящий момент листы принципиальных схем.

Default Power Object Names

Это поле используется для предварительного задания имен портов питания.

OrCAD Load Options

Выпадающий список **Copy Footprint From / To** служит для определения поля библиотечного элемента системы OrCAD, которое будет использоваться для импорта или экспорта изображения элемента принципиальной схемы системы Protel. Опция **OrCAD Ports** портов препятствует изменению размера портов в редакторе схем, что полезно в случаях, когда проектируемую схему необходимо вернуть обратно в OrCAD (где не поддерживается изменение размера портов).

Default Template File

Здесь задается файл шаблона листа, используемый по умолчанию при создании нового листа принципиальной схемы.

Вкладка Graphical Editing

Clipboard Reference

Если эта опция включена, то при выполнении операций копирования (**Edit » Copy**) или вырезания (**Edit » Cut**), система будет запрашивать указание точки привязки. Это используется при копировании части схемы, которую надо будет вставить обратно в схемный лист. Указатель мыши при этом будет удерживать вставляемый блок именно за эту точку..

Add Template to Clipboard

При включении этой опции шаблон листа будет копироваться в буфер обмена при выполнении операции копирования (**Copy**) и вырезания (**Cut**).

Convert Special Strings

Включение этой опции позволяет увидеть на экране содержимое специальных строк, в том виде в котором они будут напечатаны. Подробнее использование специальных строк будет описано в подразделе *Шаблоны листов принципиальных схем* главы *Настройки редактора принципиальных схем*.

Display Printer Fonts

Так как не все шрифты поддерживаются тем или иным устройством вывода, Windows пытается автоматически подменить шрифт. Включение данной опции позволит просматривать на экране текст в том виде, в котором он будет напечатан на бумаге.

Center of Object

При включении данной опции указатель мыши удерживает перемещаемый объект в точке привязки (если объект имеет такую точку) или за его центр (если объект не имеет точки привязки, например, прямоугольник).

Object's Electrical Hot Spot

При включении данной опции указатель мыши удерживает перемещаемый объект в ближайшей горячей точке (например, конец вывода).

Auto Zoom

При включении данной опции и переходе к какому-либо элементу принципиальной схемы автоматически изменяется масштаб отображения. В противном случае масштаб остается постоянным.

Single 'V' Negation

При включении данной опции все идентификаторы цепей (порты, метки цепей, входы листа), имеющие в начале названия символ \, воспринимаются как инверсные соответствующей цепи.

Undo Stack Size

Задаёт максимальное число последовательного выполнения команды отмены предыдущего действия (**Undo**). Следует помнить, что данная команда расходует память, и неоправданно большое число может значительно затруднить работу.

Вкладка Default Primitives

На данной вкладке можно определить графические примитивы, используемые по умолчанию для всех объектов, которые могут использоваться редактором принципиальных схем. Типичные изменения, которые вы можете сделать, включают: изменение типа шрифтов, формата позиционных обозначений, формата типа элементов и т. д., а также изменение цвета отображения связей и текста. Заданные пользователем значения по умолчанию сохраняются в файле **ADVSCHE.DFT**. Система допускает создание и загрузку пользовательских DFT файлов. Для предотвращения изменения настроек по умолчанию следует включить флажок Permanent.

Параметры листов принципиальных схем

Редактор принципиальных схем предлагает широкий набор функций управления размерами и стилями листов схем, включая создание пользовательских листов и повторное использование шаблонов листов. Все эти функции позволяют пользователю легко создавать и использовать листы, настроенные для специфических устройств или целей.

Для того, чтобы определить настройки текущего листа схемы, выполните команду меню **Design » Options**. На экране появится диалоговое окно **Document Options**.

Вкладка Sheet Options

Диалоговое окно **Document Options** обеспечивает управление различными атрибутами листа, такими как стиль и размер листа, системный шрифт, шаг сетки, информация о разработчиках схемы и т. д.

Задание размеров листа принципиальной схемы

Система предлагает пользователям 10 стандартных размеров листов в дюймовой или метрической системе мер, а также возможность задания произвольного размера листа. В последнем случае максимальный размер листа не должен превышать 65x65 дюймов (1650x1650 мм).

На экране компьютера и удобнее отображать альбомную ориентацию листа, но система позволяет просматривать на экране и выводить на печать схемы как в альбомной, так и в книжной ориентации. Стандартные размеры листов в альбомной ориентации приведены в таблице 3.1.

Таблица 3.1

Размер листа	Ширина и высота в дюймах		Ширина и высота в мм	
	Ширина	Высота	Ширина	Высота
A	11.00	8.50	279	216
B	17.00	11.00	432	279
C	22.00	17.00	559	432
D	34.00	22.00	864	559
E	44.00	34.00	1078	864
A4	11.69	8.27	297	210
A3	16.54	11.69	420	297
A2	23.39	16.54	594	420
A1	33.07	23.39	840	594
A0	46.80	33.07	1188	840
ORCAD A	9.90	7.90	251	200
ORCAD B	15.40	9.90	391	251
ORCAD C	20.60	15.60	523	396
ORCAD D	32.60	20.60	828	523
ORCAD E	42.80	32.80	1087	833
LETTER	11.00	8.50	279	216
LEGAL	14.00	8.50	356	216
TABLOID	17.00	11.00	432	279

Максимально доступная рабочая область на листе (со скрытыми границами) будет зависеть от устройства вывода. Многие принтеры и плоттеры не могут печатать на краю листа, поэтому может возникнуть необходимость сделать пробные распечатки, чтобы определить точную доступную рабочую область. Из-за этого при использовании таких устройств нельзя применить спецификации (технические требования)

ANSI и ISO на границы. Редактор принципиальных схем имеет возможность исправить эту ситуацию, за счет изменения масштаба листа при выводе на принтер или плоттер.

В случае, если отображаются границы листа, доступная рабочая область будет еще меньше. Заданная по умолчанию граница листа уменьшает рабочую область на 0.2 — 0.4 дюйма (примерно 5 — 10 мм), в зависимости от выбранного размера листа.

Template

Здесь отображается имя шаблона, на котором базируется данный лист принципиальной схемы. Более подробно вопросы, связанные с созданием и использованием шаблонов листов схем, будут рассмотрены позднее в подразделе *Шаблоны листов принципиальных схем*.

Show Borders

При включении границ листа, необходимо учитывать, что не все устройства могут печатать непосредственно на краю листа бумаги. Например, лазерные принтеры обычно резервируют поля примерно 0.15 дюймов (4 мм) вокруг области печати. Поэтому невозможно, при использовании стандартных размеров листов, например, А или А4, и при печати в масштабе 100% включить в распечатку стандартные границы. Поэтому, чтобы задействовать максимальную область печати принтера придется изменить масштаб печати.

Title block

Protel обеспечивает два встроенных типа прорисовки основной надписи схемы. Это форматы **Standard** (собственный системы Protel) и **ANSI**. Выбор нужного формата производится в соответствующем выпадающем списке. В случае создания использования пользовательской основной надписи, например, в стандарте ЕСКД, необходимо выключить встроенную и нарисовать собственную.

Snap Grid

Определяет шаг перемещения курсора на листе принципиальных схем в сотых долях дюйма. Данную опцию рекомендуется не отключать, чтобы все элементы схемы при перемещении были выровнены, за исключение специфических случаев (размещение текста).

Visible Grid

Определяет шаг видимой сетки на листе принципиальных схем в сотых долях дюйма. Для удобства рекомендуется задавать ее кратной шагу курсора.

Electrical Grid

Электрическая сетка поддерживает функцию **Guided Wiring** редактора принципиальных схем. Определяет некоторую область вокруг связи или вывода элемента, при попадании в которую при перемещении или добавлении таковых, появляется горячая точка (**Hot Spot**), в которой можно произвести электрическое соединение.

Вкладка Organization

Для введения информации о схеме и организации, ее разработавшей, в основную надпись служит вкладка **Organization** в диалоговом окне **Document Options**. Каждое поле здесь связано со специальной строкой основной надписи. Более подробно вопросы, связанные с использованием специальных строк, будут рассмотрены позднее в подразделе *Шаблоны листов принципиальных схем*.

Единицы измерения на листах принципиальных схем

Редактор принципиальных схем и редактор библиотек элементов принципиальных схем имеют разрешение 0.01 дюйма (0.3 мм). Координаты курсора, отображаемые в строке состояния, представлены также в сотых долях дюйма.

Изменение системного шрифта

Системный шрифт используется в описании позиционных полей схем, в основной надписи, в названиях и номерах выводов и т. д. Более подробно вопросы, связанные с использованием шрифтов, рассмотрены в разделе *Объекты редактора принципиальных схем*.

Шаблоны листов принципиальных схем

Границы листа, основная надпись внедренная графика составляют то, что называется листовым шаблоном. В редакторе принципиальных схем имеется несколько таких шаблонов (по одному на определенный размер листа).

Пользователь имеет возможность самостоятельно создавать шаблоны листов схем. Создаются они как обычные схемные листы. После размещения на листе всех необходимых объектов, выберите команде меню **File » Save Copy As** для того, чтобы сохранить данный лист, как файл шаблона с расширением **.DOT**. Сохраненный шаблон можно использовать как при создании листов схем как текущего, так и нового проекта.

Помимо пользовательской основной надписи и дополнительной графики, шаблоны листов принципиальных схем могут содержать специальные строки, которые обеспечивают автоматического добавления служебной информации к документу при выводе на печать. Более подробно это описано в пункте *Создание пользовательского шаблона*.

Special Strings

Специальные строки — это текстовые строки, которые распознаются и интерпретируются редактором схем при выводе листа на печать. Каждая такая строка либо связана с каким-нибудь полем вкладки **Organization** диалогового окна **Document Options** (например, **.TITLE**), либо содержит текущую информацию (например, **.DATE**). Специальные текстовые строки могут быть помещены как в шаблон листа, так и непосредственно на лист с принципиальной схемой.

Сохранив специальные строки в шаблоне листа, отпадает необходимость в точном размещении текста каждый раз, когда пользователь начинает новый проект. Вы про-

сто вызываете диалоговое окно **Document Options** для данного листа и заполняете необходимые поля. При выводе листа на печать каждая специальная строка будет замечена текстом, введенном в соответствующее поле диалогового окна. Строка будет помещена на место символа "." (точка).

Для того, чтобы просмотреть на экране текст, содержащийся в специальных строках, не выводя схему на печать, необходимо, включить флажок **Convert Special Strings** на вкладке **Graphical Edit** диалогового окна **Preferences**, появляющемся после выполнения команды меню (**Tools » Preferences**).

Ниже приведены специальные строки, связанные с соответствующими полями диалогового окна **Document Options**.

.ORGANIZATION	Текст из поля Organization (название организации).
.ADDRESS1	Текст из первого поля Address (адрес).
.ADDRESS2	Текст из второго поля Address (адрес).
.ADDRESS3	Текст из третьего поля Address (адрес).
.ADDRESS4	Текст из четвертого поля Address (адрес).
.SHEETNUMBER	Текст из поля Sheet No. (номер листа).
.SHEETTOTAL	Текст из поля Sheet Total (листов всего).
.TITLE	Текст из поля Document Title (наименование документа).
.DOCUMENTNUMBER	Текст из поля Document No. (номер документа).
.REVISION	Текст из поля Document Revision (редакция).

Ниже приведены специальные строки, вставляющие в документ текущую информацию при выводе на печать.

.DOC_FILE_NAME	Имя файла, содержащего данную схему.
.DOC_FILE_NAME_NO_PATH	Имя листа без полного адреса.
.TIME	Текущее время.
.DATE	Текущая дата.

Для того, чтобы разместить специальную строку на листе, необходимо выполнить команду меню **Place » Annotation** и ввести название строки, включая точку.

Создание пользовательского шаблона

Шаблоны документов системы Protel 99 SE хранятся в базе данных проекта по адресу **Program Files\Design Explorer 99 SE\System\Templates.ddb**. Пользователь может создавать и добавлять в эту базу данных свои шаблоны, создавать собственные базы данных шаблонов или хранить шаблоны непосредственно в проекте

Для создания пользовательского шаблона необходимо выполнить следующие операции:

1. Откройте базу данных проекта и папку, где будет храниться создаваемый шаблон. Выполните команду **File » New** для создания нового пустого листа.
2. Выполните команду **Design » Options**, и щелкните мышкой на вкладке **Sheet Options**.

3. Выберите размер листа в выпадающем списке **Standard styles**.
4. Выключите флажок **Title Block** и нажмите кнопку **OK**.

Обратите внимание, стандартная основная надпись исчезла. Приблизьте правый нижний угол листа, чтобы начать формировать пользовательскую основную надпись (приближение наиболее удобно делать с помощью горячей клавиши: переместите курсор на нужное место и нажмите клавишу **PageUP**).

Теперь вы готовы для создания новой основной надписи:

5. Выберите инструмент для рисования графических линий на панели **Drawing Tools** или выполните команду меню **Place » Drawing Tools » Line**. Курсор примет вид перекрестия.
6. Перед тем, как нарисовать линию нажмите клавишу **Tab**, после чего откроется диалоговое окно **Line dialog** задания атрибутов линии.
7. В этом окне щелкните по полю **Color box** и в окне **Color Selector** выберите цвет 4 (черный). Нажмите кнопку **OK**, чтобы закрыть окно выбора цвета.
8. После возвращения в диалоговое окно **Line dialog** в списке **Line Width** установите наименьшую (**Smallest**) ширину линии.
9. Нажмите кнопку **OK**, чтобы принять все изменения.
10. Поместите курсором в нужном месте листа для того, чтобы начать рисование первого сегмента ломаной линий контура основной надписи.
11. Переместите курсор в место, где вы хотели бы закончить первый сегмент ломаной и щелкните левой кнопкой мыши. Повторяйте эти действия, пока не возникнет необходимость рисования новой ломаной линии.
12. Щелкните правой кнопкой мыши (или нажмите клавишу **ESC**) для завершения рисования текущей ломаной линии. Переместите курсор в другое и нарисуйте новую ломаную линию.
13. Теперь вы готовы к размещению текста в основной надписи.
14. Выполните команду меню **Place » Annotation**.
15. Перед размещением текстового поля нажмите клавишу **Tab** для изменения атрибутов текста.
16. Нажмите кнопку изменения шрифта **Font Change**.
17. В поле размера наберите число 16 и щелкните **OK**.
18. В поле текста наберите **.TITLE** и щелкните на кнопке **OK**, чтобы закрыть диалоговое окно. Отметим, что в первой позиции должен обязательно стоять символ точки.
19. Расположите курсор в соответствующей графе основной надписи и нажмите левую кнопку мыши. Временно отключите сетку размещения нажав и удерживая клавишу **CTRL**.

20. Нажмите клавишу **TAB** для повторного вызова диалогового окна **Annotation**. Введите специальную строку **.DOCUMENTNUMBER** и выберите подходящий размер шрифта. Наведите курсор на то место, где вы хотите разместить строку и щелкните левой кнопкой мыши. Аналогичным образом расположите другие специальные строки в соответствующих местах блока основной надписи.
21. Нажмите **ESC** для выхода из текущей команды.
22. С помощью пункта меню **Place » Drawing Tools » Graphic** можно включить в шаблон любое графическое изображение. Для этого в диалоговом окне **Image file** необходимо выбрать нужный графический файл и нажать кнопку **OK**. Далее указателем мыши задается местоположение графического объекта. Следует помнить, что при вставке графического файла сам файл в проект не вставляется, а вставляется только ссылка на него, поэтому при переносе проекта графическое изображение может быть потеряно.

Теперь можно сохранять эту страницу как шаблон:

23. Выберите из меню пункт **File » Save As**.
24. Введите имя файла и установите тип файла **Schematic template binary (*.dot)**.
25. Расширение **.DOT** определяет данный файл как шаблон листа принципиальной схемы, который можно использовать новых или уже существующих проектов.

Задание шаблона, используемого по умолчанию

Существует возможность определить шаблон, который будет использоваться при создании нового файла по умолчанию. Для этого необходимо воспользоваться пунктом меню **Tools » Preferences** и в окне **Preferences** в нижней части вкладки **Schematic** нажать кнопку **Browse**, после чего выбрать базу данных с необходимым шаблоном и сам шаблон. Если нужная база данных отсутствует в выпадающем списке, нажмите кнопку **Add**, найдите ее и нажмите кнопку **OK**.

После нажатия кнопки **OK** имя выбранного шаблона появится в поле **Default Template File**. Теперь при создании проекта с помощью команды меню **File » New** новый лист будет использовать выбранный шаблон. Заметим, что формирующие шаблон объекты, теперь не могут быть отредактированы, и какие-либо изменения можно сделать только над самим шаблоном.

Обновление существующих шаблонов

Шаблоны могут применяться как к активным, так и ко всем открытым листам в любое время. В меню **Design** присутствуют три пункта для работы с шаблонами:

- Опция **Update Current Template** (обновить текущий шаблон) — применяется для обновления листов, использующих измененный шаблон. После изменения шаблона будет задан вопрос: необходимо ли обновлять все открытые файлы. Отрицательный ответ означает обновление только активного листа.
- Опция **Set Template File Name** (задать имя файла шаблона) — удаляет существующий шаблон и заменяет его на выбранный.

- Опция **Remove Template** (удалить шаблон) — удаляет шаблон, но сохраняет размер листа из старого шаблона.

Работа с редактором принципиальных схем

Сервер редактора принципиальных схем системы Protel 99 SE включает два редактора документов: Schematic Sheet Editor (редактор принципиальных схем) и Schematic Library Editor (редактор библиотек компонентов схем). Работа в этих редакторах представляет собой размещение объектов на листе, но, не смотря на то, что стратегия размещения, позиционирования, удаления объектов, редактирования их атрибутов и т.д. является общей для обоих редакторов, цели и решаемые ими задачи различны.

В редакторе принципиальных схем создается, редактируется и проверяется принципиальная схема разрабатываемого устройства, а в редакторе библиотек аналогичная работа производится над компонентами и библиотеками компонентов для схемы.

Изменение масштаба просмотра листа

Каждый вновь открываемый лист появляется на отдельной вкладке в окне просмотра проекта. Существует возможность приближения (**zoom in**) или отдаления (**zoom out**) любого листа. Меню **View** содержит несколько пунктов, управляющих видом активного документа, включающих **Fit All Objects** (максимальный обзор, при котором видны все объекты) и **Fit Document** (на экране отображается весь лист). Система имеет горячие клавиши для изменения вида:

- PageUp — приближение
- PageDown — удаление
- Home — центрирование экрана относительно положения курсора
- End — обновление экрана

Перемещение в редакторе принципиальных схем

Панорамирование редактора принципиальных схем

Если при изменении масштаба просмотра весь лист схемы на экран не помещается, то появятся стандартные для системы Windows линейки прокрутки, с помощью которых можно перемещаться по листу как с мелким, так и с крупным шагом.

Кроме линеек прокрутки существует еще один способ перемещения по листу — автоматическое панорамирование (autorpanning). Автопанорамирование доступно всякий раз, когда выполняются операции по редактированию схемы, например, размещение, выбор, перемещение или удаление объектов. Курсор при этом имеет форму перекрестия.

Такой курсор можно переместить либо обычным образом посредством мыши, либо с помощью клавиш со стрелками на клавиатуре. При приближении курсора к краю

окна происходит автоматическое перемещение вида. Этот процесс и называется автопанорамированием. Для увеличения скорости перемещения необходимо удерживать клавишу **SHIFT**. В диалоговом окне **Preferences (Tools » Preferences)** на вкладке **Graphical Editing** расположен движок **Speed**, определяющий повышенную скорость перемещения вида при автопанорамировании. На этой же странице можно выбрать стиль автопанорамирования — **Fixed Size Jump** (скачок фиксированного размера), при котором перемещение осуществляется с установленным шагом или **ReCenter**, при котором происходит сдвиг на расстояние, равное половине окна, а затем производится центрирование вида относительно положения курсора.

Просмотр принципиальных схем

Когда проект состоит из большого количества листов расположение объектов становится трудоемкой задачей, требующей значительных временных затрат. Для облегчения этого процесса в редактор схем включены мощные средства просмотра.

Для того, чтобы просмотреть объекты, расположенные на текущем листе или во всем проекте, необходимо на вкладке **Browse** в поле режима просмотра установить значение **Primitives** (примитивы).

Ниже можно выбрать любой тип объектов, доступных в редакторе принципиальных схем, а также любой элемент данного типа (рис. 3.3).

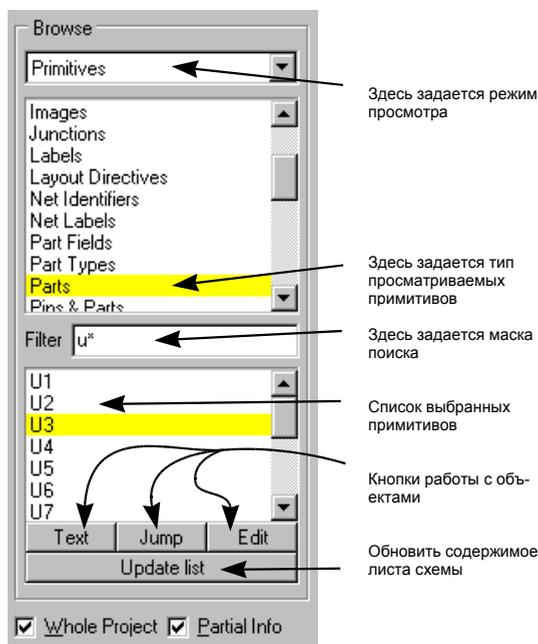


Рис. 3.3. Просмотр графических примитивов с помощью панели управления.

Нажатием кнопок **Text**, **Jump** и **Edit** можно выполнять с выбранным элементом различные действия.

- Нажатие на кнопку **Jump** вызовет переход к выбранному объекту, причем он будет расположен в центре экрана.
- Нажатие на кнопку **Text** приведет к такому же результату, но при этом, если у объекта есть текстовое поле, появится диалоговое окно редактирования текста.
- Нажатие на кнопку **Edit** вызовет те же действия, но кроме того появится диалоговое окно редактирования атрибутов объекта.
- С помощью кнопки **Update List** можно обновить список просмотра.
- При выборе какого-либо элемента из списка информация о нем появляется в строке состояния.

Переходы в редакторе принципиальных схем

Переходы представляют собой другой способ быстрого перемещения по листу принципиальной схемы. Поддерживаются следующие виды переходов: в начало координат, в заданную точку, к одной из десяти меток и к метке ошибки **Error Marker**, что позволяет избавить пользователя от необходимости постоянного изменения масштаба при работе над проектом, особенно достаточно большой схеме.

Переход в начало координат

Переход в начало координат осуществляется командой **Jump » Origin**. Началом координат в редакторе принципиальных схем считается нижний левый угол листа, а в редакторе библиотек компонентов схем — центр листа.

Переход в заданную точку по координатам

Эта опция позволяет переместиться в нужную точку листа, набрав на клавиатуре соответствующие координаты.

Использование позиционных меток

Редактор принципиальных схем позволяет использовать до десяти пользовательских позиционных меток, с быстрым доступом в любую из них через нажатие всего двух клавиш. Маркеры расставляются с помощью команды меню **Edit » Set Location Marks**.

Переход к позиционной метке

Переход к позиционной метке осуществляется командой меню **Edit » Jump » Location Mark X** или последовательным нажатием горячих клавиш **J-1**, **J-2** и т.д.

Использование окна просмотра проекта для перемещения по проекту

Окно просмотра проекта системы Design Explorer отображает на экране все документы, относящиеся к базе данных проекта, а также логическую связь между различными листами схемы.

Работа с окном просмотра дерева проекта аналогична работе в Проводнике Windows, где данные также представлены в виде иерархического дерева. Нажатие на значки "+" или "-" открывает или закрывает ветви дерева проекта, показывая или скрывая

находящиеся в нем листы. Щелчок мыши по определенному листу открывает его, а сам лист становится активным документом в окне работы с проектом.

Другим способом просмотра документов проекта является применение кнопок переключения по иерархии (**Up/Down Hierarchy**), расположенных на главной панели инструментов.

Размещение объектов на принципиальной схеме

Процесс размещения и манипулирования объектами в редакторе принципиальных схем достаточно прост и сходен с работой в других графических редакторах — сначала выбирается тот или иной инструмент, щелчком мыши определяется его местоположение, затем мышь перемещается в другую точку и т.д.

За подробной информацией по размещению объектов разного типа обратитесь к подразделу *Электрические примитивы* раздела *Объекты редактора принципиальных схем*.

Любой объект можно разместить на листе схемы с помощью команд меню, панели инструментов или горячих клавиш. Ниже приводится пример размещения проводника. В разделе *Библиотеки элементов принципиальных схем* описан процесс размещения компонентов.

Рисование проводников

Чтобы разместить проводник на текущем листе необходимо выполнить следующие действия:

1. Приблизьте лист, для чего нажимайте клавишу **PageUp** до тех пор, пока сетка не станет хорошо различима.
2. Выберите пункт меню **Place » Wire** (горячие клавиши: **P, W** или нажмите кнопку **Wire** на панели инструментов **Wiring Tools**).

Заметьте, что на конце стрелки курсора появилось перекрестие. Такой вид курсор приобретает при выполнении каких-либо операций редактирования (изменить стиль курсора можно с помощью команды меню **Tools » Preferences**).

3. Нажмите левую кнопку мыши (или клавишу **Enter**) для определения положения начальной точки проводника.

Заметьте, что перемещение курсора привязано к сетке некоторой сетке **Snap Grid**, которая может отличаться от обычной сетки на листе. Шаг сетки устанавливается командой **Design » Options**.

4. Потяните сегмент проводника в нужном направлении. Нажмите левую кнопку мыши (или клавишу **Enter**) для установки конечной точки текущего сегмента.

◆ Для удаления последнего угла (вершины) при размещении проводника нажмите клавишу BACKSPACE .

5. Переместите курсор в новую точку и опять щелкните левой кнопкой мыши. Появится второй сегмент, началом которого является конец предыдущего. Повторяйте эту операцию пока проводник не будет полностью нарисован.

6. Для завершения создания проводника нажмите правую кнопку мыши.

Заметьте, что форма курсора осталась прежней. Это говорит о том, что все еще установлен режим размещения проводника и можно начинать рисование нового проводника без обращения к команде меню **Place » Wire**.

7. Для выхода из режима размещения нажмите клавишу **ESC** (или повторно правую кнопку мыши). Перекрестие на конце указателя мыши должно пропасть.

Различные режимы ручного и автоматического размещения проводников

Редактор принципиальных схем имеет шесть режимов размещения шин и проводников. Переключение между режимами осуществляется с помощью клавиши пробел. Существует два ортогональных режима, режим произвольного угла и автоматический режим.

Режим произвольного угла (Any Angle Mode)

Позволяет располагать проводники под любым углом.

Ортогональный режим 90/90 (90/90 Line Mode)

В этом режиме проводники располагаются только вертикально или горизонтально с изломом кратным 90°. Здесь существуют еще два вспомогательных режима в зависимости от того, какой сегмент проводника (длинный или короткий) должен примыкать к указателю мыши.

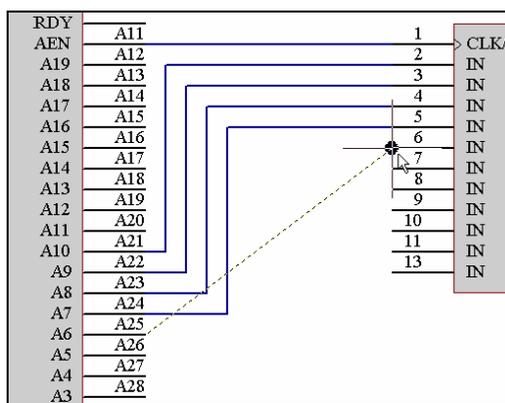


Рис. 3.4. При включенном режиме Auto Wire после задания начала проводника в точке вывода A25 и его окончания в точке 6 система автоматически нарисует проводник.

Ортогональный режим 45/90 (45/90 Line Mode)

В этом режиме углы между сегментами проводников выбираются кратными 45° (то есть из ряда $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$). Сегмент, непосредственно примыкающий к указателю мыши, может располагаться либо под углом 0° , либо под углом в 45° .

Режим автоматического размещения (Auto Wire Mode)

Режим автоматического размещения проводников позволяет ортогонально соединять две произвольно выбранные точки на листе принципиальной схемы. Изменение параметров режима производится в соответствующем окне, появляющемся после нажатия клавиши **Tab**. О том, что вы работаете в режиме **Auto Wire**, пользователю подсказывает наличие пунктирной линии (рис. 3.4).

Автоматизация прорисовки проводников

Для ускорения процесса размещения соединительных проводников в редакторе принципиальных схем применены две эффективные функции: автоматическая прорисовка проводников и автоматическая вставка узловой точки (рис. 3.5).

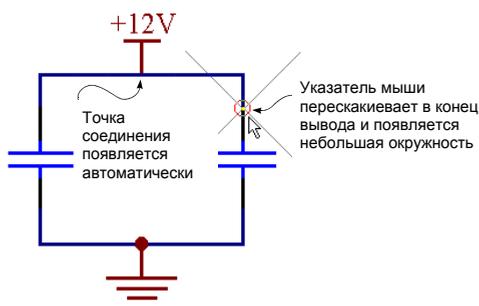


Рис. 3.5. Автоматизация прорисовки проводников.

Когда размещаемый проводник попадает в поле действия электрической сетки какого-либо электрического объекта (например, вывода или другого проводника) указатель мыши фиксируется на этом объекте и появляется "горячая" точка в виде подсвеченного кружка. Эта точка показывает, в каком месте можно сделать правильное соединение.

Данная функция значительно ускоряет процесс размещения проводника и избавляет от необходимости использовать операцию увеличения масштаба. Как только курсор попадает в нужную точку для завершения операции достаточно щелкнуть кнопкой мыши. Если в этом месте необходима узловая точка, она будет добавлена автоматически. Функция вставки узловой точки помещает ее в случае Т-образного соединения двух проводников, а также в случае ортогонального подсоединения проводника к выводу или порту питания.

♦ Делайте шаг электрической сетки всегда несколько меньшим, чем шаг видимой сетки, в противном случае поиск электрических объектов будет затруднен.

Создание двумерных графических объектов

Двухмерные графические объекты, например, прямоугольники и полигоны, создаются и размещаются на схеме аналогично проводникам, шинам и линиям (рис. 3.6). Каждый объект требует разного количества щелчков мыши для его определения. Дополнительную информацию по этой теме можно найти в подразделе *Электрические примитивы* раздела *Объекты редактора принципиальных схем*.

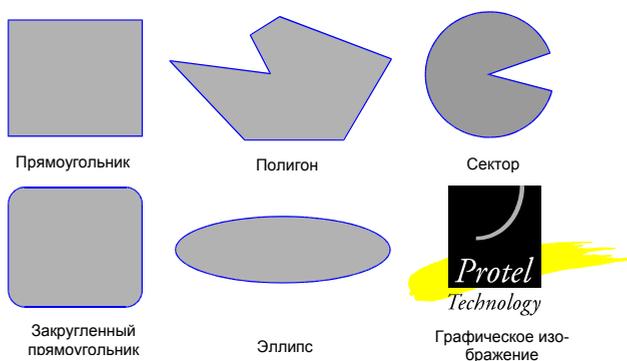


Рис. 3.6. Различные типы графических объектов редактора схем.

Обратите внимание, что в файле вместе со схемой сохраняются не сами графические изображения, а лишь ссылки на файлы, содержащие эти изображения. Поэтому при переносе проекта в другую папку необходима корректировка ссылок, что можно сделать, выполнив двойной щелчок левой кнопкой мыши на нужном объекте.

Редактирование объектов на принципиальных схемах

В редакторе принципиальных схем существует два способа редактирования объектов: изменение их атрибутов с помощью диалогового окна **Change** и изменение объектов непосредственно на листе. Некоторые операции (например, изменение цвета примитива) могут быть выполнены только посредством редактирования атрибутов, другие (например, изменение размера прямоугольника) обоими указанными способами.

Графическое редактирование объекта проще всего выполнить непосредственно на листе, для чего сначала его нужно выделить. Для редактирования объекта через окно диалога необходимо выбрать пункт меню **Edit » Change** и затем щелкнуть по нужному объекту.

Редактирование в процессе размещения объекта

Пользователь имеет возможность редактировать атрибуты объекта при размещении его на листе принципиальной схемы. Для этого, когда объект удерживается указателем мыши, нужно нажать клавишу **TAB**, после чего появится соответствующее диалоговое окно, в котором нужно сделать все необходимые изменения. У такого подхода есть ряд преимуществ:

- Изменения, сделанные во время размещения, можно сохранить как настройки по умолчанию для объектов данного типа. Все изменения сохраняются в специальном файле **ADVSCH.DFT**. Заметим, что этот метод изменения настроек по умолчанию зависит от состояния флага **Permanent** в диалоговом окне **Preferences (Tools » Preferences)**. В случае, если этот флаг установлен, то сделанные изменения отразятся только на выделенном объекте.
- По мере размещения численные идентификаторы в таких объектах, как позиционное обозначение элемента, автоматически увеличиваются на единицу.
- Отпадает необходимость в последующем редактировании объекта на листе схемы, что также ускоряет процесс проектирования.

Редактирование объектов на листе принципиальной схемы

Для изменения уже размещенного объекта нужно выполнить команду меню **Edit » Change**, поместить курсор на объект и щелкнуть левой кнопкой мыши (либо просто выполнить двойной щелчок левой кнопкой мыши на объекте). Каждый объект имеет собственный набор редактируемых атрибутов. Изменения могут быть сделаны применительно как к одному выбранному объекту, так и глобально ко всем объектам проекта.

Редактирование текста непосредственно на листе

Текстовые строки могут быть отредактированы непосредственно на листе принципиальной схемы. Для этого сначала надо выполнить один щелчок левой кнопкой мыши на текстовой надписи, а затем второй щелчок (не двойной щелчок). Данная функция работает при включении опции **Enable In-place Editing** диалогового окна **Preferences**.

Графическое редактирование — выделение фокусом и комплексное выделение

Одним из основных преимуществ графической среды редактирования является способность внести изменения в объект непосредственно на экране. Начиная процесс графического редактирования отдельного объекта или группы объектов необходимо их выделить. Для этого существует два способа — выделение фокусом и комплексное выделение.

В большинстве приложений операционной системы Windows выделение является единственным способом выбора объектов для их последующего изменения. Типичным примером является выделение одного или более объектов для копирования в буфер обмена и последующей вставки в другое место рабочего пространства. Как правило модифицируются непосредственно выделенные объекты. Например, в гра-

фических редакторах таким образом можно переместить объекты или изменить их размер.

В отличие от других приложений Windows, редактор принципиальных схем системы Protel использует два независимых метода для решения описанных задач. Оба этих метода, выделение фокусом и комплексное выделение, неоднократно применяются при создании и редактировании схемы. Комплексное выделение позволяет пользователю выполнять более сложные операции модифицирования объектов, которое было бы затруднено или невозможно при использовании метода описанного выше традиционного выделения.

Выделение фокусом

Если в редакторе принципиальных схем навести курсор на какой-либо объект и щелкнуть на нем левой кнопкой мыши, то объект станет активным, а его вид на экране изменится. Такой метод выделения объекта называется выделение фокусом. Аналогичным образом пользователь делает активным одно из открытых окон приложений Windows.

В любой момент времени выделенным таким образом может быть только один объект, при этом он отображается на экране с особыми графическими элементами — метками-манипуляторами (или рамкой выделения). Например, если щелкнуть левой кнопкой мыши по прямоугольнику, в каждом из его углов и в середине каждой из его сторон появится метка, позволяющая изменять его размеры. Если аналогичную процедуру проделать с меткой цепи, то вокруг нее появится рамка выделения (рис. 3.7).

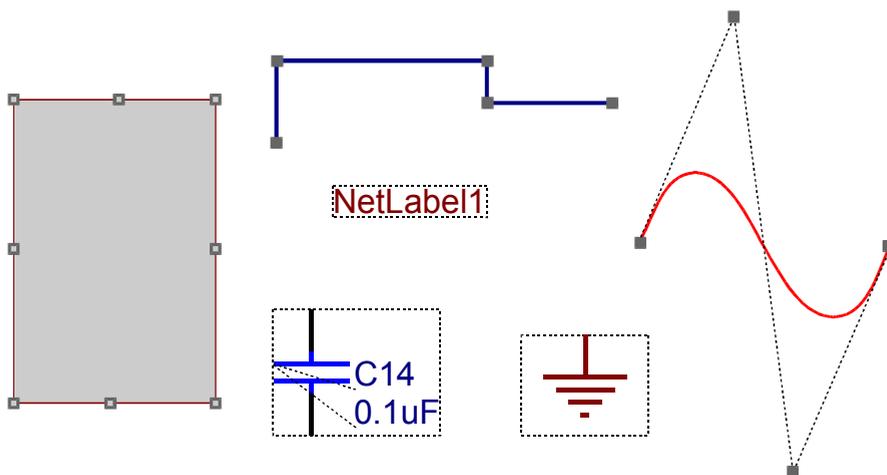


Рис. 3.7. Вид различных объектов, выделенных фокусом.

Графическое редактирование объекта, выделенного фокусом

После выделения объекта фокусом можно перемещать его в другое место листа схемы или изменять его графические характеристики.

Для того, чтобы изменить выделенный фокусом объект непосредственно на листе, нужно щелкнуть по одной из его меток-манипуляторов, после чего эта метка "приклеется" к указателю мыши. Теперь для перемещения объекта на новое место достаточно просто передвинуть туда курсор и повторно щелкнуть кнопкой мыши.

Итак, для перемещения объекта, находящегося в фокусе, нужно щелкнуть по нему мышью, для удаления такого объекта необходимо нажать клавишу **DELETE**. Более подробная методика перемещения объектов описана в подразделе *Перемещение и перетаскивание* данной главы.

Редактирование полилиний

Полилинии представляют собой объекты, имеющие специальные свойства для графического редактирования. Например, при размещении проводников, шин или графических линий каждый раз, когда изменяется направление необходимо определить их точки излома или вершины. При выделении фокусом такого объекта вершины приобретают вид специальных меток-манипуляторов. И хотя такие объекты могут иметь весьма сложную форму, действия над ними (перемещение, вырезание, копирование, вставка, удаление) идентичны действиям, производимым над простыми объектами.

Для уже прорисованных полилиний также существует специальная функция добавления и удаления вершин.

Добавление вершины к полилинии

Чтобы добавить вершину у существующей полилинии нужно:

1. Для выделения фокусом нужной полилинии щелкните на ней левой кнопкой мыши.
2. Наведите указатель мыши на сегмент полилинии в месте, где вы собираетесь добавить новую вершину. Щелкните левой кнопкой мыши и, удерживая ее, нажмите клавишу **INSERT**. В точке расположения курсора сегмент приобретет излом.
3. Переместите новую вершину в нужное место и отпустите левую кнопку мыши. Вершина создана.

Если курсор перепрыгивает в ближайшую метку-манипулятор, а не остается там, где был выполнен щелчок мыши, необходимо отключить опцию **Object's Electrical Hot Spot** в диалоговом окне **Preferences**.

Удаление вершины

Иногда при изменении формы полилинии необходимо удалить одну из вершин. Для этого нужно выполнить следующие действия:

1. Для выделения фокусом нужной полилинии щелкните на ней левой кнопкой мыши.

2. Наведите курсор на метку в вершине полилинии, которую вы хотите удалить, щелчком захватите ее и удерживайте левую кнопку мыши.
3. Для удаления этой вершины нажмите клавишу **DELETE**.

Курсор перепрыгнет в ближайшую существующую вершину.

Резюме

Как показано в этих примерах, для того, чтобы модифицировать объект непосредственно на листе, его необходимо выделить фокусом. Отметим тот факт, что над выделенным фокусом объектом, нельзя совершать операции с применением буфера обмена — копирование, вырезание, вставку, удаление. Эти функции используются только по отношению к комплексно выделенным объектам.

Комплексное выделение

Другим методом манипулирования объектами является комплексное выделение. В отличие от выделения фокусом, этот метод может быть использован по отношению как к отдельным объектам, так к группам объектов.

При комплексном выделении объектов маркеры-манипуляторы не появляются. Вместо этого объект подсвечивается цветом выделения (**selection color**), который задается командой меню **Tools » Preferences**.

Будучи выделенным таким способом, объект может быть перемещен, скопирован, вырезан или вставлен в другое окно или другое место текущего окна, а также удален. Специальные опции команды **Edit » Move** позволяет перемещать или вращать выделенные объекты посредством нажатия "горячих" клавиш. Метод комплексного выделения дает возможность использовать присутствующие в редакторе принципиальных схем функции глобального редактирования, причем возможно применение их как к выделенным, так и невыбранным объектам (обратное выделение).

Основная особенность метода комплексного выделения в редакторе принципиальных схем — это возможность выполнения различных операций, не являющихся операциями над выделенными объектами, но требующими использования левой кнопки мыши, без снятия выделения с объектов.

Выборка объектов производится следующими способами:

- непосредственное выделение, используя клавишу **SHIFT** и левую кнопку мыши для добавления (или удаления) отдельных элементов в (из) текущее выделение;
- с помощью окна охвата;
- командами меню **Edit » Select** и **Edit » DeSelect**;
- с использованием флага **Selection** в диалоговом окне редактирования атрибутов объекта. Эта опция позволяет использовать в редакторе принципиальных схем функцию глобального редактирования для изменения статуса выделения различных объектов проекта. Подробнее функции глобального редактирования описаны в этой главе в подразделе *Глобальное редактирование*.

При выполнении неправильного действия пользователь может восстановить предыдущее состояние с помощью команды меню **Edit » Undo**.

Следует помнить, что выделенные объекты остаются таковыми до тех пор, пока над ними не будет произведена операция снятия выделения с помощью команды меню **Edit » DeSelect** (горячие клавиши — последовательное нажатие клавиш **X, A**).

Непосредственное выделение отдельных объектов

Непосредственное выделение — наиболее гибкий и удобный способ выделения отдельных объектов. Ниже приведены действия по выделению объекта:

1. Наведите курсор на объект и, удерживая клавишу **SHIFT**, нажмите левую кнопку мыши.

Объект будет обновлен и прорисован цветом выделения (для изменения цвета используйте команду меню **Tools » Preferences**). Повторяя эти действия, можно добавлять другие объекты к текущему выделению.

Если выдается звуковой сигнал или нужный объект не выделяется необходимо приблизить лист схемы (нажать клавишу **PAGEUP**) и убедиться, что указатель мыши находится непосредственно над объектом. При выделении компонента электрической схемы можно щелкнуть в любом месте его условного графического обозначения. Операция выделения компонентов, а особенно составных компонентов, может занять некоторое время.

Для добавления других объектов к текущему выделению:

2. Удерживая клавишу **SHIFT**, нажмите левую кнопку мыши над добавляемым объектом.

Для удаления элемента:

3. Удерживая клавишу **SHIFT**, нажмите левую кнопку мыши над объектом, который нужно удалить из группы выделенных.

После того, как над объектом проведена операция снятия выделения, он перерисовывается своим первоначальным цветом. Выделенные объекты остаются таковыми до тех пор, пока над ними не будет произведена операция снятия выделения либо с помощью одновременного нажатия клавиши **SHIFT** и левой кнопки мыши, либо с помощью команды меню **Edit » DeSelect** (горячие клавиши — последовательное нажатие **X, A**).

Непосредственное выделения в окне

Для выделения объектов с помощью окна охвата необходимо:

1. Поместить указатель мыши в место, где нет объектов.
2. Нажать и удерживать левую кнопку мыши.

В строке состояния появится подсказка — **Choose Second Corner** (выбери второй угол).

3. Переместить мышь по диагонали, определив прямоугольную область (окно) выделения, и отпустить кнопку мыши.

Будут выделены и перерисованы текущим цветом выделения только те объекты, которые полностью попадают в отмеченный прямоугольник.

4. Для расширения группы выделения нужно повторить все предыдущие действия.

Использование меню для выполнения операции выделения

Команда меню **Edit** » **Select** позволяет выделить все объекты внутри или снаружи какой-либо области, либо вообще все объекты. Существует возможность выделения всех проводников и идентификаторов цепей, соединенных с выбранной цепью на текущем листе, посредством опции **Net** и выделения всех проводников и идентификаторов цепей, физически соединенных с выбранной цепью на текущем листе, посредством опции **Connection**.

Команда меню **Edit** » **DeSelect** имеет такие же опции, за исключением опций **Net** и **Connection**. Для вызова всплывающего меню **DeSelect** существует горячая клавиша **X**.

Команда меню **Edit** » **Toggle Selection** работает таким же образом, как одновременное нажатие клавиши **SHIFT** и левой кнопки мыши при непосредственном выделении, то есть меняет статус выделения объекта на противоположный.

Работа с выделенными объектами

Выделенные объекты могут быть вырезаны (**Cut**) или скопированы (**Copy**) в буфер обмена, и уже оттуда вставлены (**Paste**) в другой лист принципиальной схемы или в другое приложение, поддерживающее операции с буфером обмена. Удалить выделенные объекты можно с помощью пункта меню **Edit** » **Clear** или посредством одновременного нажатия клавиш **CTRL** и **DEL**.

Использование буфера обмена в редакторе принципиальных схем системы Protel ничем не отличается от других приложений Windows: выделение объектов, вырезание или копирование их в буфер обмена, вставка содержимого буфера в нужное место.

Некоторые замечания по использованию буфера обмена

- Редактор принципиальных схем имеет опцию задания точки привязки (**Clipboard Reference Location**) для операций вырезания или копирования в буфер обмена. При последующей вставке содержимое буфера обмена фиксируется в этой точке, что позволяет правильно и точно расположить вставляемые объекты. Если эта опция включена (команда меню **Tools** » **Preferences**), то при выполнении операций вырезания или копирования будет выдаваться запрос "**Choose Clipboard Reference Location**" для определения местоположения точки привязки.
- Команда **Edit** » **Copy** выполняет копирование выделенных объектов в буфер обмена.
- Команда **Edit** » **Cut** выполняет удаление выделенных объектов с листа схемы и копирование их в буфер обмена.
- Команда **Edit** » **Paste** позволяет вставить содержимое буфера обмена в активный лист.

- Перед выполнением операций вырезания или копирования необходимо убедиться, что выделенными являются только нужные объекты. Поэтому перед началом выделения необходимо выполнить полную отмену выделения командой **Edit » DeSelect » All** или последовательным нажатием горячих клавиш **X, A**.
- Для добавления объекта к текущему выделению выполните одновременное нажатие клавиши **SHIFT** и левой кнопки мыши.
- Любая операция копирования или вырезания обновляет содержимое буфера обмена, а значит, он будет содержать только последнее выделение.
- Команда **Edit » Clear** удаляет выделенные объекты без помещения их в буфер обмена (горячая клавиша **CTRL+DELETE**).
- Редактор принципиальных схем имеет опцию для включения шаблона листа в копируемый блок. Для включения этой опции нужно использовать команду меню **Tools » Preferences**.

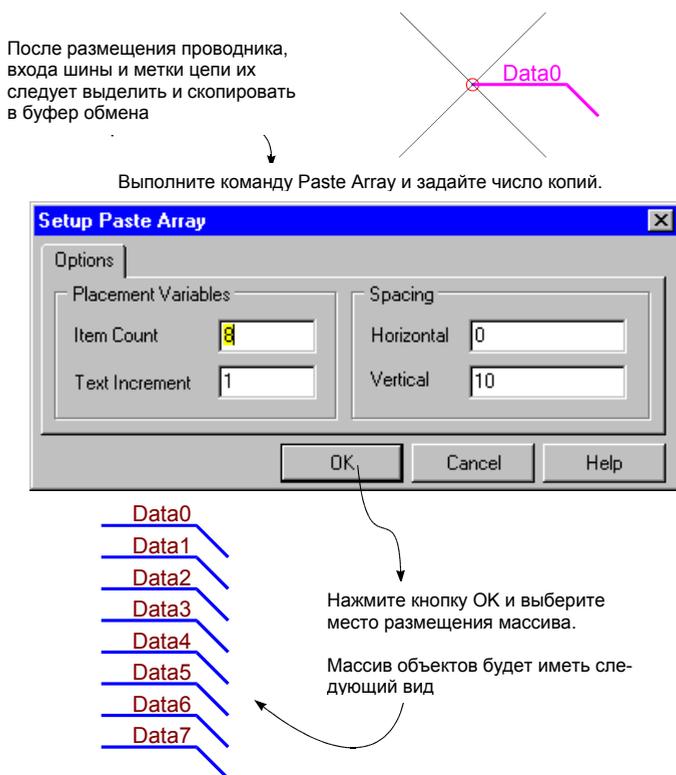


Рис. 3.8. Вставка восьми проводников с помощью функции Paste Array

Вставка упорядоченных массивов объектов

При использовании команды **Edit » Cut** (или **Edit » Copy**) в буфер обмена помещается копия текущего выделения. Команда **Edit » Paste Array** позволяет разместить на листе схемы несколько копий содержимого буфера обмена. На приведенном ниже рисунке показан пример использования функции вставки упорядоченного массива для создания набора из восьми линий данных. Функция задания точки привязки **Clipboard Reference**, описанная в предыдущем разделе, облегчает задачу создания и размещения массива элементов.

Item Count

Определяет количество копий. Например, ввод в этом окне цифры 4 приведет к размещению четырех копий содержимого буфера обмена.

Text Increment

Данный параметр используется, если нужно, чтобы численная составляющая текстовой надписи автоматически увеличивалась на заданную величину. В случае, когда таким текстом является метка цепи, установка значения 1 приведет к нумерации, например U1, U2, U3 и т. д.

Horizontal

Определяет расстояние по горизонтали между вставляемыми копиями. Установка здесь значения 10 приведет к размещению копий с шагом в 10 минимальных значений или 0.1 дюйма.

Vertical

Определяет расстояние по вертикали между вставляемыми копиями. Установка здесь значения 10 приведет к размещению копий с шагом в 10 минимальных значений или 0.1 дюйма.

Глобальное редактирование

Пользователь имеет возможность редактировать атрибуты не только одиночного объекта, но и всех объектов такого же типа на текущей схеме или, по необходимости, во всем проекте.

Пользователь может задавать признаки, по которым будут выбираться объекты для редактирования. Например, изменениям могут быть подвергнуты только те объекты, которые входят в текущее выделение, либо, наоборот, только те, которые не входят в него, либо все объекты, невзирая на их статус выделения. Если необходимо, можно создать более сложный набор признаков для функции глобального редактирования.

Фактически любой из атрибутов редактирования может быть задействован в той или иной ситуации. Простейшие примеры: изменение цвета всех сегментов проводников, связанных с определенной цепью или изменение шрифта всех обозначений цепи. Все эти атрибуты могут быть использованы при глобальном редактировании. Возможные вариации применения метода глобальных изменений ограничены только воображением разработчика.

Большое количество опций глобального редактирования могут поначалу показаться сложными. Однако, научившись их эффективно применять, пользователь получит в

свои руки важный инструмент, который избавит его от большого объема работ по ручному редактированию при разработке схемы.

Глобальное редактирование во всем проекте

Функции глобального редактирования поддерживаются также и в многолистных иерархических проектах. Эта особенность позволяет изменять элементы, расположенные в различных частях проекта, или изменять стиль оформления всех открытых в данный момент листов проекта. Установкой значения **Change Scope** можно контролировать, на какие листы будет влиять глобальное редактирование.

Стратегия глобального редактирования

Если набор признаков глобального редактирования, которые присутствуют в диалоговых окнах разных объектов, зависит от типа этих объектов, то используемая стратегия всегда неизменна. Это определение описывает общий подход к глобальному редактированию.

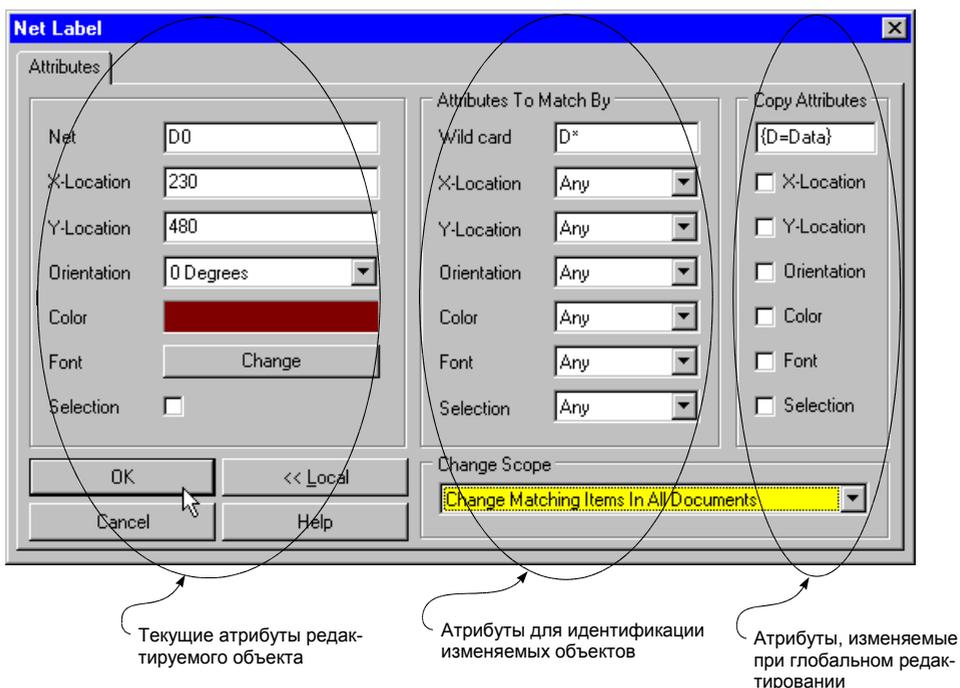


Рис. 3.9. Использование функции глобального редактирования.

Current Attributes

С помощью двойного щелчка мыши по объекту можно открыть диалоговое окно, содержащее текущие значения или установки атрибутов данного объекта. Естественно, что эти значения можно менять.

На рисунке 3.9 показано, каким образом можно изменить обозначения цепей с **D1**, **D2** и т. д. на **Data1**, **Data2** и т. д. Так как это простая строковая замена нет необходимости изменять что-либо в **Current Attributes**.

Attributes to Match By

После изменения атрибутов нажмите кнопку **Global**. Диалоговое окно расширится как показано на рисунке. В центре окна расположен столбец с заголовком **Attributes to Match By**, где определяется, каким образом идентифицировать другие объекты, к которым нужно применить сделанные здесь изменения.

Столбец **Attributes to Match By** содержит либо выпадающие списки, либо текстовые поля, в которые можно ввести текст с клавиатуры.

Выпадающие списки имеют три состояния: **Same** (применить глобальное изменение к объектам, у которых значение этого атрибута совпадает с текущим), **Different** (применить глобальное изменение к объектам, у которых значение этого атрибута не совпадает с текущим) и **Any** (состояние этого атрибута не влияет на выбор объекта при глобальном изменении). Состояние **Any** является состоянием по умолчанию.

Если все атрибуты в этом столбце установлены в **Any**, а текстовые поля содержат символ-заменитель "*", то глобальные изменения будут применены ко всем объектам этого типа.

На рисунке же продемонстрирован пример выбора обозначений цепей, начинающихся с буквы "D".

Copy Attributes

Третий столбец в диалоговом окне называется **Copy Attributes**. Этот столбец содержит либо поля флагов для каждого атрибута, либо текстовое поле, в которое можно ввести текст с клавиатуры.

В этом столбце определяется, какие атрибуты нужно изменить и, если присутствует текстовое поле, как его значение должно измениться в соответствующих объектах.

В диалоговом окне, представленном на рисунке, выполняется строковая замена {D=Data}. Синтаксис операции строковой замены (**String Substitution**) описан ниже.

Change Scope

Последним устанавливаемым в этом окне параметром является **Change Scope** (область действия изменений). Соответствующий список имеет три состояния: **the current item** (текущий элемент), **all matching items in this document** (все подходящие элементы в текущем документе) и **all matching items in the documents that make up the project** (все подходящие элементы в документах, входящих в текущий проект). Другие открытые документы, не являющиеся частью проекта, в расчет не принимаются.

Нажатие на кнопку **OK** приведет к замене всех обозначений цепей, начинающихся с символа **"D"**, на строку **"Data"**, то есть **D1** станет **Data1**, **D2** станет **Data2** и т. д., а если на схеме есть цепь с обозначением **DR**, то оно изменится на **DataR**.

Если нет уверенности в правильном выборе объектов при операции глобального редактирования, тогда можно установить флажок **Selection** в столбце **Copy Attributes**. В этом случае выбранные для редактирования объекты станут выделенными и не меняют своих атрибутов. Теперь можно проверить, какие объекты вошли в выделение, и повторно применить функцию глобального редактирования, установив поле **Selection** в столбце **Attribute to Match By**.

Использование символов-заменителей при глобальном редактировании текста

Многие объекты содержат текстовые поля. Части многоэлементных компонентов, обозначения цепей, примечания (текст в одну строку), символы листа, входы листа, обычные порты и порты питания позволяют в процессе глобального редактирования использовать символы-заменители.

В редакторе принципиальных схем могут применяться два традиционных для Windows символа-заменителя: "?" и "*". Следует отметить также, что они не чувствительны к регистру написания букв.

Заметим, что команды **Edit » Find Text** и **Edit » Replace Text** применимы к различным типам объектов. Они также поддерживают символы-заменители и синтаксис операции строковой замены.

Синтаксис операции строковой замены

Для определения изменений, которые необходимо сделать в текстовых строках, существует поле **Copy**, которое можно использовать двумя способами.

Если нужно полностью изменить значение этого поля, необходимо убрать фигурные скобки и ввести новый текст.

Фигурные скобки используются когда желательно выполнить частичную замену строки. В этом случае синтаксис такой: **{старый_текст = новый_текст}**, что означает замену части строки **"старый_текст"** на строку **"новый_текст"**. В нашем примере буква **"D"** заменяется на строку **"Data"**.

Для выполнения сложных замен можно использовать несколько пар скобок, тогда в первую очередь будет сделана самая левая замена. Это очень мощный инструмент, но обращаться с ним нужно внимательно во избежание непредвиденных результатов.

По умолчанию операция замены текста не чувствительна к регистру букв. Символ **"!"** меняет это положение. Например, строка **{!Text=text}** заменит **"Text"** на **"text"**, т.е. будет учтен регистр букв.

С помощью команды **Undo** можно отменить действие предыдущей операции.

Резюме

Аккуратное использование такого мощного инструмента, как глобальное редактирование, дает почувствовать пользователю небывалую выгоду от его применения. Однако, неправильное его использование может привести к непредвиденным результатам, особенно при редактировании сложных сочетаний объектов. Поэтому перед созданием нового выделения рекомендуется выполнять операцию очистки выделения **Edit » DeSelect All** (горячие клавиши **X, A**). И не нужно забывать о спасительной функции **Undo/Redo**.

Быстрое копирование объектов

Редактор принципиальных схем имеет мощную функцию копирования атрибутов одного объекта в другой объект такого же типа. С помощью этой функции можно "превратить" объект, находящийся на указателе мыши, в объект, который уже размещен на схеме.

Допустим у вас на указателе мыши "висит" конденсатор, тогда как на самом деле Вы хотите разместить резистор. Вместо того, чтобы нажимать клавишу **ESC** для сброса конденсатора, чтобы затем найти и выбрать в библиотеке резистор, можно воспользоваться функцией быстрого копирования для замены конденсатора на резистор.

Просто расположите курсор с конденсатором над каким-нибудь резистором уже размещенном на схеме и нажмите клавишу **INSERT**, тогда вместо конденсатора на курсоре "повиснет" резистор, который "наследует" все атрибуты резистора на схеме. Теперь, если нужно, Вы можете изменить атрибуты резистора до его размещения, нажав клавишу **TAB**.

Следите за тем, чтобы при нажатии **INSERT** перекрестие курсора находилось внутри объекта, иначе функция не сработает.

Функция быстрого копирования атрибутов (также называемая "morphing") используется для "клонирования" атрибутов схемных объектов всех типов за исключением символов листа. Эта функция работает либо для размещения новых объектов, либо для перемещения ранее размещенного элемента. Она не действует при перетаскивании соединенных электрических объектов.

Поиск и замена текста

Редактор принципиальных схем позволяет находить и заменять текст на текущем листе или нескольких листах в многолистовом проекте.

Например можно переименовать все цепи шин данных **D1, D2, D3** и т.д. в **Data1, Data2, Data3** во всем проекте. Это можно было бы сделать и с помощью функции глобального редактирования, но тогда бы пришлось использовать ее отдельно для обозначений цепей, портов и входов листа.

Для того, чтобы найти и заменить текст необходимо выполнить следующие действия:

1. Выбрать пункт меню **Edit » Replace Text**.

Появится диалоговое окно **Find And Replace Text**.

2. Введите в поле **Text To Find** искомую строку (например "D*").

Эта функция поддерживает символы-заменители: "?" — для любого, но только одного символа, и "*" — для любого набора символов.

3. В поле **Replace With** введите текст, который заменит найденный.

В примере, приведенном на рисунке 3.10, символ "D" заменяется на строку "Data", а остальная часть заменяемой строки остается прежней. Это достигается использованием описанного ниже синтаксиса замены строк, использующего фигурные скобки.

4. Установите поля **Scope** и **Options**, которые описаны ниже.
5. Нажмите кнопку **OK** для выполнения замены.

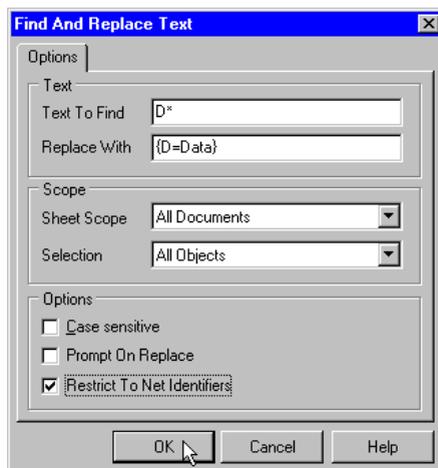


Рис. 3.10. Поиск и замена текста.

Область действия (Scope)

Команда замены текста может действовать только в текущем документе (**Current Document Only**) или во всех открытых документах (**All Open Documents**). Поиск объектов, в текстовых полях которых производится замена, может выполняться среди выделенных или, наоборот, среди невыделенных объектов.

Учет регистра (Case Sensitive)

Если установить флаг **Case Sensitive**, то операция замены будет сделана с учетом регистра букв, т.е. поиск будет производиться при точном совпадении шаблона с найденным текстом с учетом того, какими буквами заглавными или прописными введен шаблон текста. Замененный текст всегда полностью совпадает с текстом, введенным в поле **New Text**.

Подтверждение замены (Prompt on Replace)

При включении этого флага после каждого удачного поиска выдается запрос на подтверждение замены.

Ограничение действия функции замены (Restricting changes to net identifiers)

При включении этого флага функции поиска и замены текста распространяется только на идентификаторы цепей, то есть на обозначения цепей, порты питания, обычные порты и входы листа.

Сложные условия замены текста (Conditional String Substitutions)

Оба символа-заменителя ("*" и "?") могут быть использованы для определения строки поиска. Символы-заменители не чувствительны к регистру букв.

При строковой замене может применяться синтаксис `{старый_текст=новый_текст}`. Это значит, что можно изменить часть строки "старый_текст" на "новый_текст". В примере, приведенном на рисунке, буква "D" заменяется на текст "Data".

Существует возможность использования нескольких пар фигурных скобок для определения сложных замен. В таком случае, в первую очередь выполняется самая левая замена. Такую операцию нужно выполнять с большой осторожностью, так как она может привести к неожиданным результатам.

При использовании только что описанного синтаксиса можно добиться, чтобы замена осуществлялась с учетом регистра букв. Например, `{!Текст=текст}` произведет замену строки "Текст" на строку "текст".

Выравнивание объектов

Существует два метода выравнивания объектов. Первый — выравнивание группы выделенных объектов по двум осям (меню **Edit** » **Align** » **Align**). Второй – выравнивание объектов по одной из осей (рис. 3.11).

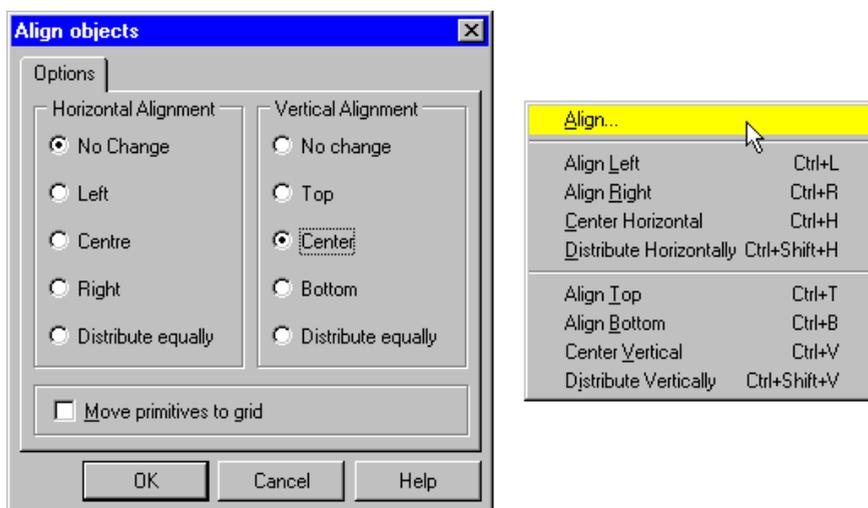


Рис. 3.11. Различные способы выравнивания объектов.

Замечания по процедуре выравнивания объектов

- Убедитесь, что только выделены требуемые объекты, для этого перед началом выделения выполните операцию очистки текущего выделения (горячие клавиши **X**, **A**).
- Выберите все элементы, которые нужно выровнять.

- Если необходимо выравнивание и по горизонтали, и по вертикали, для вызова соответствующего диалогового окна используйте команду меню **Edit » Align » Align**. Если требуется выравнивание только по одному направлению, выберите соответствующий пункт из меню **Edit » Align**.
- При включении флаг **Move primitives to Grid**, выравнивание производится с привязкой к ближайшему узлу сетки.
- Если результаты выравнивания вас не удовлетворяют, используйте команду меню **Edit » Undo**.

Перемещение и перетаскивание

В редакторе принципиальных схем системы Protel перемещение и перетаскивание представляют собой две разные операции. Перемещение (**Move**) объектов — это изменение их позиции на листе без поддержки соединений, перетаскивание (**Drag**) — это изменение позиции объектов на листе с поддержкой электрических соединений.

Перемещение и перетаскивание могут быть применены как к одиночным объектам, так и к группе выделенных объектов.

Выполнение перемещения

Для перемещение одиночного объекта необходимо расположить указатель мыши над объектом, нажать левую кнопку мыши и, удерживая ее, переместить указатель в нужное место. Для перемещения группы объектов сначала их нужно выделить. Затем произвести те же действия, как и с одиночным объектом, выбрав в качестве перемещаемого любой из выделенных объектов. Эту же операцию можно произвести с помощью команды меню **Edit » Move**.

Перемещение объектов в стеке

Графические объекты могут быть размещены на листе так, что они будут перекрывать друг друга. При создании нового объекта он располагается на передний план поверх уже существующих на листе элементов. При перемещении объекта он сохраняет свою позицию на экране, относительно других перекрывающихся элементов. Редактор принципиальных схем имеет специальную функцию для изменения "глубины залегания" какого-либо элемента, то есть для изменения порядка в стеке. Нажатие горячей клавиши **M** вызывает подменю **Move**, в котором можно выбрать один из следующих пунктов:

Move To Front

Данная команда позволяет переместить объект на передний план, также изменить его местоположение на листе. При использовании этого пункта меню пользователю предлагается выбрать объект перемещения. Необходимо щелкнуть мышью по нужному объекту, переместить его в нужное место и повторным щелчком расположить на листе.

Bring To Front

Эта команда позволяет поместить объект на передний план. При использовании этого пункта меню пользователю предлагается выбрать объект. Необходимо щелкнуть мышью по нужному объекту, после чего его положение в стеке изменится без перемещения по листу.

Send To Back

Эта команда помещает объект на задний план. При использовании этого пункта меню пользователю предлагается выбрать объект. Необходимо щелкнуть мышью по нужному объекту, после чего его положение в стеке изменится без перемещения по листу.

Bring To Front Of

Помещает объект над каким-либо элементом. При использовании этого пункта меню пользователю предлагается выбрать два элемента, после чего первый расположится над вторым без перемещения по листу.

Send To Back Of

Помещает объект под какой-либо элемент. При использовании этого пункта меню пользователю предлагается выбрать два элемента, после чего первый расположится под вторым без перемещения по листу.

Перетаскивание объектов

В случае, когда возникает необходимость в перераспределении объектов на листе схемы с сохранением всех соединений (например, при добавлении новой цепи), используется перетаскивание (**Drag**) объектов.

Перетаскивание одиночного объекта

Для перетаскивания одиночного объекта нужно расположить указатель мыши на объекте, нажать клавишу **CTRL** и, удерживая ее, нажать левую кнопку мыши. Затем отпустить клавишу **CTRL**, переместить курсор в нужное место листа и отпустить левую клавишу мыши.

Перетаскивание группы объектов

Для перетаскивания группы объектов необходимо выделить нужные объекты, выбрать команду меню **Edit » Move » Drag Selection**, щелчком мыши определить опорную точку и затем перетащить выделенные объекты в нужное место листа.

Советы по выполнению перетаскивания

- Если плотность расположения проводников высокая, то при перетаскивании они могут пересечься и при срабатывании функции автоматической установки узловых точек может произойти нежелательное соединение разных цепей. Во избежание этого рекомендуется выключать функцию **Auto-Junction** в диалоговом окне **Tools » Preferences**.

- Другая функция, помогающая выполнять перетаскивание при плотном расположении проводников, называется **Drag Orthogonal** (окно **Tools » Preferences**). Ее включение позволяет перемещать проводники под любым углом и не дает им пересекаться при перетаскивании (рис. 3.12). Затем можно изменить форму любого проводника посредством взятия его в фокус и последующих операций по перемещению, добавлению или удалению его вершин. Ранее в этой главе в пункте *Выделение фокусом* были описаны операции редактирования выделенного фокусом объекта.
- Если при перетаскивании разносятся два вывода, которые ранее соприкасались, то между ними автоматически будет добавлен проводник.
- Одновременное нажатие клавиш **CTRL** и **SPACEBAR** при перетаскивании позволяет поворачивать объекты.
- Нажатие на клавишу **SPACEBAR** при перетаскивании переключает режим **Drag Orthogonal**.

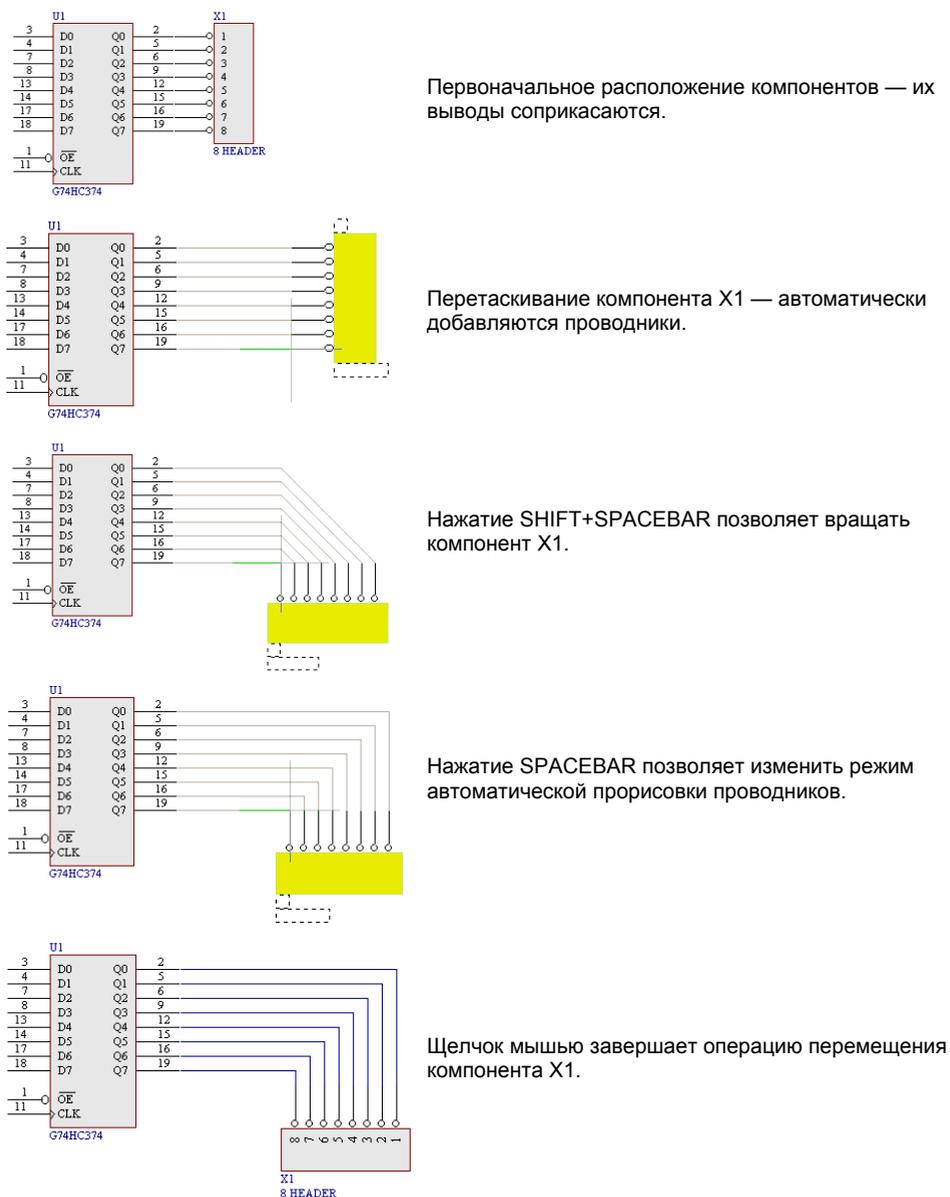


Рис. 3.12. Пример перетаскивания объектов.

Горячие клавиши редактора принципиальных схем

Вложенные команды

Редактор принципиальных схем Protel 99 SE имеет мощную функцию, которая позволяет выполнять две операции одновременно, благодаря наличию вложенных команд.

Эта функция предлагает пользователю дополнительные удобства и гибкость. Например, при размещении проводника вы вспомнили, что этот проводник нужно присоединить к еще не введенному порту. Нет необходимости покидать режим размещения, просто нужно нажать горячую клавишу для размещения порта (**P, R**), расположить порт в нужном месте, нажать на клавишу **ESC** для выхода из режима расположения порта и затем присоединить проводник к порту.

По такой схеме на фоне одной операции может быть выполнено множество других операций. Количество запусков второго (фонового) процесса во время выполнения первого процесса зависит от требований к программному обеспечению каждого из этих двух процессов. Заметим, что фоновый процесс может быть запущен только с помощью горячих клавиш.

Отмена обновления экрана

Всякий раз когда производится изменение размера и/или положения обзора экрана, содержимое рабочего пространства перерисовывается, для того, чтобы отразить эти изменения. Существует возможность прервать процесс перерисовки в любое время с помощью клавиши **SPACEBAR**. Это сохранит время, в случае когда пользователь желает немедленно сделать повторное изменение вида экрана и ему не нужен подробный вид текущего содержимого экрана.

Горячие клавиши для мыши

В системе Protel горячие клавиши могут быть как для работы с клавиатурой, так и для работы с мышью. Например, нажатие на клавиш **P, N** позволяет разместить обозначение цепи вместо того, чтобы обращаться к меню Place для выбора пункта меню **Net Label**, что приведет к тому же результату. Использование левой кнопки мыши вместо клавиши **ENTER** и правой кнопки мыши вместо клавиши **ESC** позволяет выполнять многие операции без использования клавиатуры. В разного рода диалоговых окнах наоборот удобно использовать клавиши **ENTER** и **ESC** вместо нажатий мышью на кнопки **OK** и **CANCEL**.

Для некоторых операций, где нежелательно перемещение мыши по листу схемы, работа с клавиатурой выглядит эффективнее. Например, выключение **Snap Grid** при

размещении текстовой строки (**V, G**) или изменение масштаба обзора при перемещении группы выделенных объектов.

В редакторе принципиальных схем поддерживаются стандартные горячие клавиши операционной системы Windows: **ALT-F4** для закрытия приложения, **CTRL-TAB** для переключения окон. Все остальные горячие редактора схем являются специфическими. Кроме того, имеется возможность создания пользовательских горячих клавиш, процесс определения которых описан в разделе *Изменение настроек среды проектирования Design Explorer* главы *Работа в среде Design Explorer*.

Допускается также запись какой-либо последовательности действий средствами операционной системы в виде макроса и назначение для него собственной горячей клавиши. Информацию об этом можно найти в руководстве пользователя Microsoft Windows.

Горячие клавиши для клавиатуры

Существует два способа создания горячих клавиш для клавиатуры. Первым является создание в специальном редакторе Keyboard Shortcut Editor, вызываемого с помощью команды меню **Client Menu » Customize**. Здесь создаются горячие клавиши, которые непосредственно запускают тот или иной процесс. Например, нажатие на клавиш **CTRL+T** приведет к выравниванию выделенных ранее объектов по их верхнему краю.

Какой-либо процесс может быть запущен с помощью так называемых горячих клавиш меню. Ряд команд меню содержат подменю, которые открываются при их выборе, другие команды подменю не имеют и запускают соответствующую процедуру. Например, нажатие на клавишу **P** вызывает меню **Place**, а последующее нажатие на **N** позволяет разместить на листе метку цепи. Нажатие на клавишу **T** вызывает меню **Tools**, а последующее нажатие на **N** вызовет диалоговое окно **Netlist Creation**.

Горячие клавиши для клавиатуры имеют более высокий приоритет, чем горячие клавиши меню, поэтому при совпадении комбинаций будут срабатывать именно клавиатурные.

Горячие клавиши вызова команд меню:

A	Edit » Align (выравнивание)
B	View » Toolbars (панель инструментов)
E	Edit (редактирование)
F	File (файл)
H	Help (помощь)
J	Edit » Jump (переход)
	Edit » Set Location Marks (установка позиционных меток)
M	Edit » Move (перемещение)
D	Design (проект)
P	Place (размещение)
R	Reports (отчеты)

S	Edit » Select (выделение)
T	Tools (инструменты)
V	View (обзор)
W	Window (окно)
X	Edit » DeSelect (отмена выделения)
Z	Zoom (масштаб)

Горячие клавиши для клавиатуры:

CTRL+BACKSPACE	Redo (повторить)
ALT+BACKSPACE	Undo (отменить)
PGUP	Zoom In (увеличить масштаб)
CTRL+PGDN	Zoom All (показать всю схему)
PGDN	Zoom Out (уменьшить масштаб)
END	Redraw (перерисовать схему)
CTRL+HOME	Jump to Origin (перейти в начало координат)
HOME	Pan (перейти к позиции указателя мыши)
SHIFT+←	перемещение курсора влево (на 10 шагов сетки)
←	перемещение курсора влево (на 1 шаг сетки)
SHIFT+↑	перемещение курсора вверх (на 10 шагов сетки)
↑	перемещение курсора вверх (на 1 шаг сетки)
SHIFT+→	перемещение курсора вправо (на 10 шагов)
→	перемещение курсора вправо (на 1 шаг сетки)
SHIFT+↓	перемещение курсора вниз (на 10 шагов сетки)
↓	перемещение курсора вниз (на 1 шаг сетки)
SHIFT+INSERT	Paste (вставить)
CTRL+INSERT	Copy (копировать)
SHIFT+DELETE	Cut (вырезать)
CTRL+DELETE	Clear (очистить)
DELETE	Удаление выделенного объекта
CTRL	Временное отключение сетки размещения
CTRL+1	Масштаб 100%
CTRL+2	Масштаб 200%
CTRL+4	Масштаб 400%
CTRL+5	Масштаб 50%
CTRL+F	Find Text (найти текст)
CTRL+G	Find And Replace Text (найти и заменить в тексте)
CTRL+V	Горизонтальное выравнивание объектов по их центрам
CTRL+B	Горизонтальное выравнивание объектов по их нижним краям
CTRL+T	Горизонтальное выравнивание объектов по их верхним краям
CTRL+SHIFT+H	Расставить объекты по горизонтали на одинаковом расстоянии

CTRL+H	Вертикальное выравнивание объектов по их центрам
CTRL+L	Вертикальное выравнивание объектов по их левому краю
CTRL+R	Вертикальное выравнивание объектов по их правому краю
CTRL+SHIFT+V	Расставить объекты по вертикали на одинаковом расстоянии
F1	Вызов контекстной справки
F3	Поиск следующего фрагмента текста
SHIFT+F4	Расположить все открытые окна мозаикой
SHIFT+F5	Расположить все открытые окна каскадом
SHIFT+CTRL+Left-Click	Переместить одиночный объект
SHIFT+Left-Click	Вкл/выкл выделение одиночного объекта
CTRL+Left-Click	Перетащить одиночный объект
Left-Click	Выделить объект фокусом
Left-Double-Click	Изменить объект
CTRL+Left-Hold-Down	Перетащить одиночный объект
ALT	Переместить объект только по оси Y
ALT+SHIFT	Переместить объект только по оси X
Left-Hold-Down	Переместить объект / группу объектов

Наиболее часто используемые горячие клавиши

- SPACEBAR — прервать перерисовку экрана
- X, A — снять выделение объектов
- V, D — изменить масштаб так, чтобы лист умещался целиком на экране
- V, F to — изменить масштаб так, чтобы на экране умещались все объекты
- PAGE UP — увеличить масштаб изображения
- PAGE DOWN — уменьшить масштаб изображения
- HOME — центровать экран относительно положения курсора
- END — обновить экран
- TAB — вызвать окно редактирования атрибутов объекта, непосредственно перед его размещением
- SPACEBAR — перевернуть объект на 90° перед его размещением
- X — отразить объект зеркально относительно оси Y перед его размещением
- Y — отразить объект зеркально относительно оси X перед его размещением
- BACKSPACE — удалить последнюю вершину (шины, линии, полигона) перед размещением

- SPACEBAR — переключиться на следующий режим размещения проводника (шины, линии, полигона) перед размещением
- ESC — отменить режим, в котором находится пользователь
- CTRL — временно отключить сетку размещения snap grid перед размещением объекта
- CTRL+TAB — перейти к другому открытому документу в среде проектирования Design Explorer
- ALT+TAB — перейти к другому открытому окну в среде Windows
- F1 — если курсор имеет форму перекрестия, то вызывается список горячих клавиш.

Операции Undo и Redo

Редактор принципиальных схем имеет многоуровневый механизм отмены и возврата к состоянию, предшествующему операции отмены. Каждое действие сохраняется в специальной области памяти по стековому принципу. При выполнении команды Undo из стека выбирается последняя операция.

Редактор схем оптимизирован таким образом, чтобы операции **Undo** и **Redo** обеспечивали минимальный расход памяти, однако, существует возможность очистить стек этих команд, временно присвоив в диалоговом окне **Preferences** параметру **Stack Size** (размер стека) значение равное нулю.

Объекты редактора принципиальных схем

Среда проектирования и при разработке принципиальной схемы, и при работе в редакторе библиотек состоит из двух основных компонентов: объектов, размещаемых на листе и являющихся строительным материалом любой разработки, и процедур, выполняемых системой или пользователем для создания, изменения, сохранения объектов, а также для составления отчетов.

В редакторе принципиальных схем существует два типа объектов: Прimitives и Составные объекты. Прimitives — это базовые схемные элементы, включающие как электрические объекты (проводники, точки соединения, порты питания, метки цепей и входы листов), так и графические объекты (текст, линии). Составные объекты, например, условные графические обозначения компонентов схемы, строятся из групп электрических и графических объектов (прimitives).

Каждый объект имеет набор собственных атрибутов, которые могут быть определены индивидуально. В следующем разделе данной главы приведено короткое описание всех primitives.

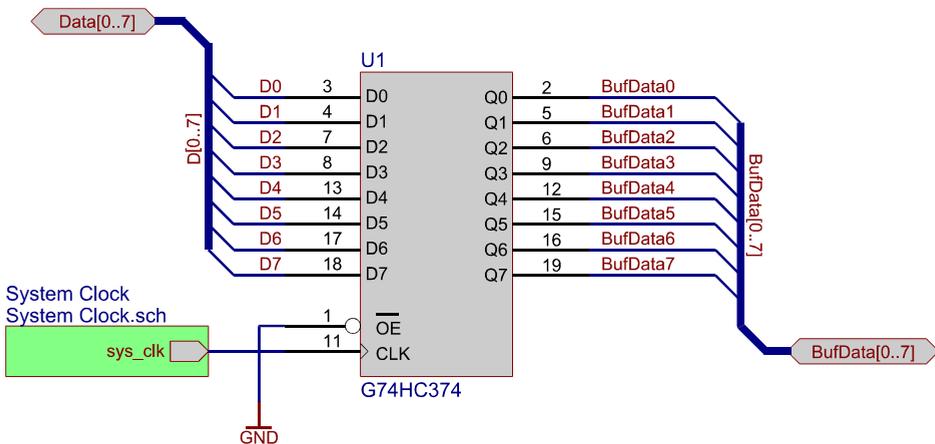


Рис. 3.13. Построение проекта заключается в упорядоченном расположении различных графических объектов на схеме.

Электрические примитивы



Wire (проводник)

Проводники — ломаные линии, размещаемые на листе принципиальной схемы и обозначающие электрические соединения.



Junction (точка соединения)

Точка соединения — маленький объект в виде окружности, используемый для обозначения логического соединения пересекающихся проводников на схеме.



Bus Entry (вход шины)

Вход шины — специальный символ, расположенный под углом 45°, который используется для обозначения соединения проводника и шины. Вход шины позволяет графически соединять две различные цепи в одной точке шины (смотри рисунок). Если бы это было сделано с помощью простых проводников, то в точке образовалось бы электрическое соединение.



Bus (шина)

Шина — специальный графический объект, обозначающий сразу несколько проводников. Шины не имеют электрических параметров, поэтому они идентифицируются с помощью меток цепей и портов, как показано на рисунке. Нумерация проводов в шине может быть как возрастающей ($D[0..7]$), так и убывающей ($D[7..0]$).



Net Label (метка цепи)

При соединении проводником нескольких компонентов образуется цепь (**Net**). При создании списка цепей каждой цепи присваивается уникальный идентификатор.

Вместо того, чтобы позволить редактору принципиальных схем присвоить какой-либо цепи сгенерированный им самим идентификатор, можно вручную определить метку этой цепи. Метки цепей могут располагаться в вертикальном или горизонтальном положении, но так, чтобы их ниж-

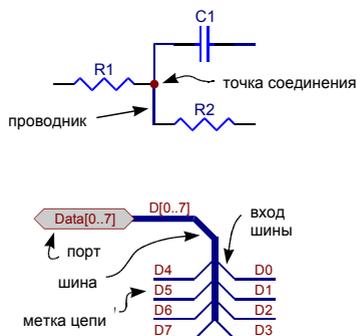


Рис. 3.14. Основные электрические примитивы.

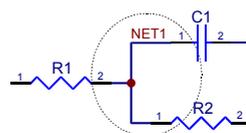


Рис. 3.15. Метка цепи.

ний левый угол примыкал к проводнику или шине, с которыми они связаны. На рисунке NET1 имеет три вывода (узла): R1-2, R2-1 и C1-1.

Power Port (порт питания)



Порт питания — специальный символ, обозначающий цепь питания (или заземления). Порты питания позволяют в удобной форме показывать цепи питания (и земли) в любом месте проекта, которые затем можно подсоединить к выводам или проводникам. Порт питания всегда идентифицируется названием цепи, но не его графическим изображением, что позволяет избежать лишних ошибок.

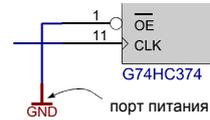


Рис. 3.16. Порт питания

Port (порт)



Порты используются для соединения цепей на разных листах проекта, при этом порты имеют одинаковые имена. Порты также могут соединять подчиненные листы с входами листов (**Sheet Entries**) на символах листов в проектах со сложной иерархией. Для поворота портов при размещении используется клавиша **SPACEBAR**.

Sheet Symbol (символ листа)



Символы листа представляют собой графическое обозначение другого (как правило, подчиненного) листа схемы. Связь между символами листа и самим листом, определяется атрибутом **File Name** (имя файла), который должен быть таким же, как имя данного подчиненного листа.

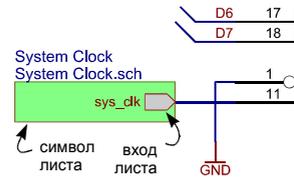


Рис. 3.17. Символ и вход листа.

Sheet Entry (вход листа)



Вход листа создает электрическое соединение между примыкающей к нему цепью на главном листе, и портом с таким же именем, находящемся на подчиненном листе. Изменять положение входа листа внутри символа листа можно перетаскиванием с помощью мыши.

Probe Directive (контрольная точка)



Контрольные точки представляют собой специальные маркеры, размещаемые на листе принципиальной схемы и определяющие узлы, в которых при численном моделировании будет определена форма электрических сигналов.

Test Vector Directive (тестовый вектор)

Тестовые векторы — это специальные символы, используемые для определения узлов с тестовыми векторами сигналов моделирования. Тестовые векторы обозначаются номером, который определяет соответствующий столбец файла тестового вектора, используемого при моделировании.

Stimulus Directive (вход сигнала)



Вход сигнала представляют собой символы, определяющие узлы или цепи, на которые при численном моделировании подаются заданные входные сигналы.

PCB Layout Directive (топологическая директива)

Топологические директивы дают возможность для какой-либо цепи описать правила для разводки ее на печатной плате.

No ERC Directive (директива отмены проверки правил электрических соединений)



Эти специальные символы позволяют исключить те или иные выводы из процедуры проверки правил электрических соединений (ERC).

Pin (вывод)



Выводы — специальные объекты, которые имеют электрические характеристики и используются для подачи (вывода) сигналов в (из) элементы электрических принципиальных схем. Подробная информация по работе с выводами приведена в главе Редактор Библиотек.

Графические (не электрические) примитивы

Графические примитивы, не имеющие электрического смысла, используются для нанесения на лист схемы вспомогательной информации. С их помощью можно создавать пользовательские рамки листа и основные надписи, добавлять в схемы разного рода замечания и инструкции.

Line (линия)



Линия (полилиния) — это графический объект с произвольным числом соединенных вместе сегментов.

Polygon (полигон)



Полигон — графический объект в виде многоугольника. Определение полигона требует как минимум трех щелчков мыши. Каждая вершина полигона может быть перемещена независимо от других вершин.

Elliptical Arc (эллиптическая дуга)



Эллиптические дуги используются для создания незамкнутых круговых или эллиптических кривых. Для задания эллиптической дуги необходимо сделать пять щелчков мыши: задать центр, радиус по оси X, радиус по оси Y, первую и вторую конечные точки.



Bezier Curve (кривая Безье)

Этот инструмент используется для создания линий искривленной формы (например отрезка синусоиды или меандра). Для определения кривой Безье необходимо по крайней мере четыре щелчка мыши. Щелчки, превышающие четвертый, создают новую кривую, примыкающую к предыдущей.



Text Annotation (текстовое примечание)

Примечание — это одна строка текста, используемая для размещения на листе схемы заметки или комментария. Этот текст может содержать заголовки частей проекта, информацию о версиях, дате и времени или любой другой текст описательного или рекомендательного характера.



Text Frame (текстовое поле)

Текстовые поля могут содержать несколько строк любого текста и используются для размещения в рабочем пространстве более детальных заметок или примечаний.



Rectangle (прямоугольник)

Прямоугольники бывают закрашенные или не закрашенные графические. Для задания прямоугольника необходимо два щелчка мыши: в верхнем левом и нижнем правом углах.



Rounded Rectangle (прямоугольник с закругленными углами)

Задается также, как и предыдущий элемент.



Ellipse (эллипс)

Эллипсы бывают закрашенные или не закрашенные. Для определения эллипса необходимо тремя щелчками мыши обозначить его центр, радиус по оси X и радиус по оси Y.



Pie (круговой сектор)

Круговые секторы могут быть закрашенные или не закрашенные. Секторы определяется четырьмя щелчками мыши: центр круга, его радиус, начальная и конечная точки.



Graphic Image (графическое изображение)

В проект может быть включено графическое изображение. Поддерживаются следующие форматы файлов: BMP, TIF, JPG, WMF.

Заметим, что копия изображения вместе с листом схемы не сохраняется, сохраняется только лишь ссылка на графический файл. Поэтому, при изменении местоположения файла, ссылку на него в проекте необходимо обновить с помощью двойного щелчка на изображении.

В подразделе *Размещение объектов на принципиальной схеме* раздела *Работа с редактором принципиальных схем* приведены примеры размещения объектов при проектировании схемы.

Блок процедуры

Редактор принципиальных схем позволяет создавать блоки процедур, содержащих подробную информацию о конфигурации параметров этой процедуры.

В блоке процедуры можно сохранять настройки специальных процессов проекта. Например, проект может иметь специальные настройки для механизма проверки электрических правил (ERC), которые можно сохранить вместе с проектом для последующего их использования.

Конфигурирование блока процедуры

После создания пустого блока процедуры для его конфигурирования необходимо произвести на нем двойной щелчок мыши.

Имя процедуры (Process Name)

Кнопка **Browse** используется для задания процедуры, который будет запускаться из контейнера.

Кнопка **Configure**

Если блок запускает одну из процедур сервера обработки списка соединений, то он может быть сконфигурирован посредством кнопки **Configure**. Нажатие на эту кнопку вызывает соответствующее диалоговое окно, где можно произвести необходимые настройки, и, нажав кнопку **OK**, вернуться снова в окно **Process Container** (аналогичным образом может быть сконфигурирована процедура **Sch:Annotate**).

Кнопка **Change**

После первого запуска процедуры из соответствующего блока появляется возможность проверить ее параметры. Это можно осуществить с помощью кнопки **Change**. Параметры можно изменить либо в открывшемся окне, либо в окне, вызываемом кнопкой **Configure**.

Кнопка **Change** введена потому, что не все процедуры могут быть сконфигурированы из соответствующего блока (нажатие на кнопку **Configure** приведет к запуску такой

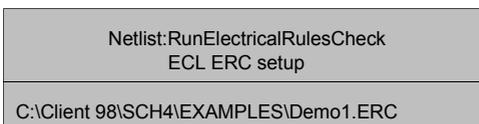


Рис. 3.18. Блок процедуры.

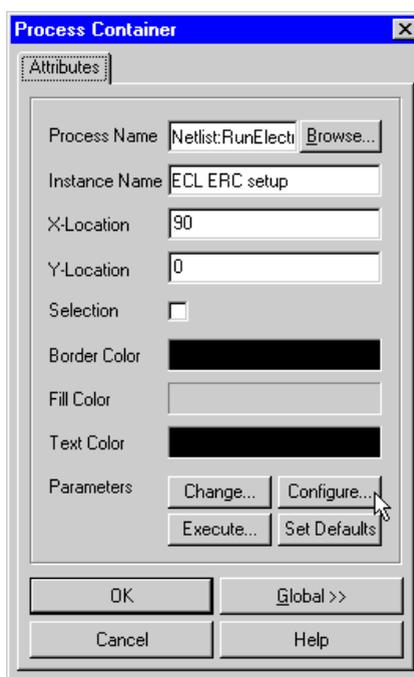


Рис. 3.19. Редактирование параметров блока процедуры.

процедуры). Для настройки параметров этих процедур нажмите кнопку **Change**, после чего введите необходимые параметры (по одному в каждой строке).

Выполнение

Для запуска процедуры из соответствующего блока нужно выполнить команду меню **Tools » Process Container » Run** или нажать кнопку **Execute** в диалоговом окне редактирования блока.

Файлы, создаваемые блоками процедур

Если в результате работы процедуры из соответствующего блока генерируется какой-либо файл (например, процедуры **Netlist:CreateNetlist**), то в диалоговом окне **Change Parameters** необходимо задать имя этого файла. Такой файл открывается при запуске процесса и присоединяется к проекту в качестве подчиненного листа в окне просмотра дерева проекта.

Шрифты

Редактор принципиальных схем поддерживает шрифты TrueType, которые поставляются с операционной системой Windows, обеспечивает различное их начертание и масштабирование. Масштабируемые шрифты PostScript и немасштабируемые растровые шрифты Windows могут быть использованы, если они являются частью файла векторного изображения, импортированного в лист схемы.

Организация шрифтов

Формально все шрифты можно разделить на две группы по принципу их использования теми или иными объектами:

Индивидуально настраиваемые шрифты используются в обозначениях компонентов, метках цепей, текстовых примечаниях, рамках с текстом, именах символов листа и именах файлов символов листа.

Системные шрифты используются в тексте рамки листа, основной надписи, именах выводов, номерах выводов, портах, портах питания и входах листа.

Изменение шрифтов

Для того, чтобы определить принадлежность шрифта к первой группе, т.е. к индивидуально редактируемым шрифтам, нужно сделать двойной щелчок левой кнопкой мыши по тексту. Появится диалоговое окно, которое в случае индивидуально редактируемого шрифта будет иметь кнопку **Font Change**, нажатие на которую в свою очередь вызывает диалоговое окно **Font**. Если двойной щелчок не приводит в вызову окна или нет кнопки **Font Change**, тогда этот текст использует системный шрифт.

Существует возможность определить шрифт, используемый по умолчанию, для всех объектов, имеющих настраиваемый шрифт. Сделать это можно в диалоговом окне **Preferences** (команда меню **Tools » Preferences**) на вкладке **Defaults**, где выбирается объект, для которого нужно установить шрифт (например, метка цепи), и нажимается кнопка **Edit Values**. Далее необходимо обычным образом выбрать нужный шрифт. Теперь при начертании всех меток цепи будет использован выбранный шрифт. Заме-

тим, что ранее расположенные на листах схемы метки цепи не изменят свой вид, их начертание можно изменить, применив функцию глобального редактирования.

Настройка системного шрифта

Изменение системного шрифта осуществляется с помощью команды меню **Design » Options**, которая вызывает окно **Document Options**. В этом окне имеется вкладка **Sheet Options**, на которой расположена кнопка **Change System Font**. После нажатия на эту кнопку появляется диалоговое окно **Font**, где можно выбрать системный шрифт.

Библиотеки элементов принципиальных схем

Для эффективного использования библиотек элементов принципиальных схем, а также управления ими, необходимо понимание связей между библиотеками, компонентами и частями составных компонентов (секциями). Библиотеки представляют собой это базы данных, в которых хранятся описания компонентов электрических схем. В свою очередь каждый компонент состоит из одной или более секций, например, микросхема 74SL00 состоит из четырех секций, конденсатор состоит из одной секции, а электромагнитное реле может быть представлено как компонент, состоящий из катушки индуктивности и контактной группы.

Компоненты создаются и редактируются в редакторе библиотек элементов принципиальных схем (Schematic Library Editor), независимом редакторе документов, который включает в себя инструменты, необходимые для управления библиотеками. Этот редактор может работать одновременно с редактором принципиальных схем, а среда проектирования Design Explorer обеспечивает все необходимые связи между разрабатываемыми листами принципиальных схем и задействованными библиотеками. Например, вы можете переместить символ секции компонента непосредственно на лист схемы, а затем редактировать информацию об этом компоненте внутри исходной библиотеки.

Что такое библиотека элементов принципиальных схем?

Система Protel 99 SE поставляется с обширными библиотеками компонентов, условные графические обозначения которых выполнены как в общепринятом стандарте, так и стандартах ANSI-IEEE и DeMorgan.

Библиотеки принципиальных схем состоят из описаний компонентов, которые включают в себя отдельные секции, размещаемые на листах принципиальных схем. Компоненты могут иметь одну или несколько секций, например, микросхема 74LS00.

Редактор библиотек элементов принципиальных схем позволяет управлять библиотечными базами данных, создавать новые компоненты, копировать и перемещать их между библиотеками. Для получения более подробной информации об использовании этого редактора обратитесь к подразделу *Редактор библиотек элементов принципиальных схем*.

Что такое компонент и секция?

Под понятием "компонент" подразумевается какой-либо физический элемент, размещаемый на разрабатываемой печатной плате, например, резистор, интегральная схема или разъем. Многие электронные компоненты включают в себя несколько отдельных секций. Таким компонентом является, например, микросхема 74SL00, состоящая из четырех одинаковых секций, или сдвоенный операционный усилитель.

Где находятся библиотеки компонентов принципиальных схем?

В системе Protel 99 SE библиотеки элементов принципиальных схем содержатся в специальных базах данных. Большинство из этих баз данных ориентированы на конкретных производителей, то есть компания разрабатывает единые библиотеки для микросхем того или иного изготовителя. Также существуют специальные базы данных, например, библиотеки графических элементов для проектирования устройств на основе ПЛИС или схемотехнического моделирования.

Библиотечные базы данных системы Protel 99 SE находятся в папке по адресу `\Program Files\Design Explorer 99 SE\Library\Sch`.

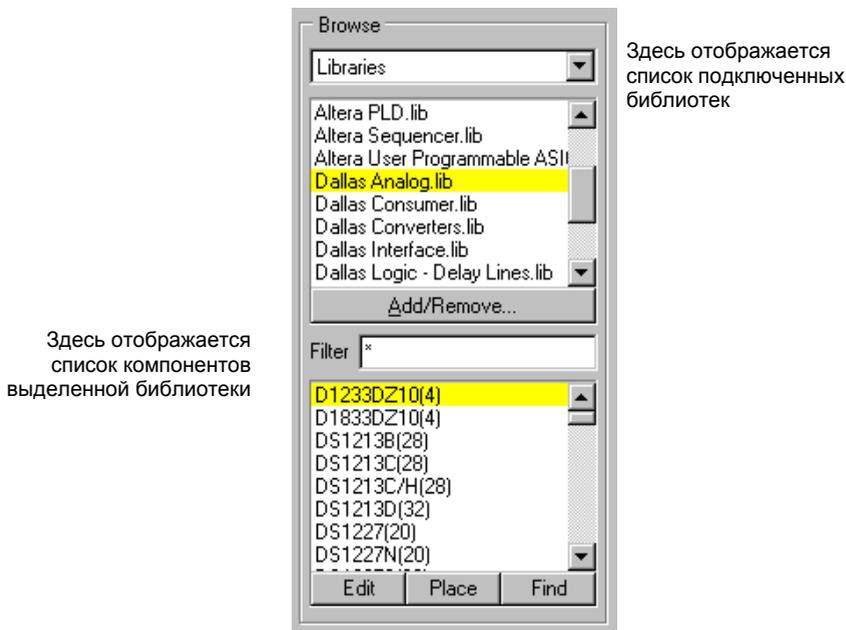


Рис. 3.20. Просмотр библиотек с помощью панели управления редактора схем.

Доступ к компонентам, необходимым для конструирования

Чтобы получить доступ к компонентам той или иной библиотеки, она должна быть добавлена в список доступных библиотек редактора принципиальных схем, который отображается в верхней части его панели управления (рис. 3.20).

Существует два способа добавить или удалить библиотеку: нажатием кнопки **Add/Remove** на соответствующей панели или с помощью команды меню **Design » Add/Remove Library**. Появится диалоговое окно **Change Library List**, в котором производится подключение новых библиотек или отключение уже подключенных библиотек из списка **Current File List**. Максимальное количество библиотек, одновременно подключенных к системе, ограничивается только ресурсами используемого компьютера.

Библиотеки подключаются к системе только один раз, после чего их компоненты могут быть использованы для создания различных принципиальных схем.

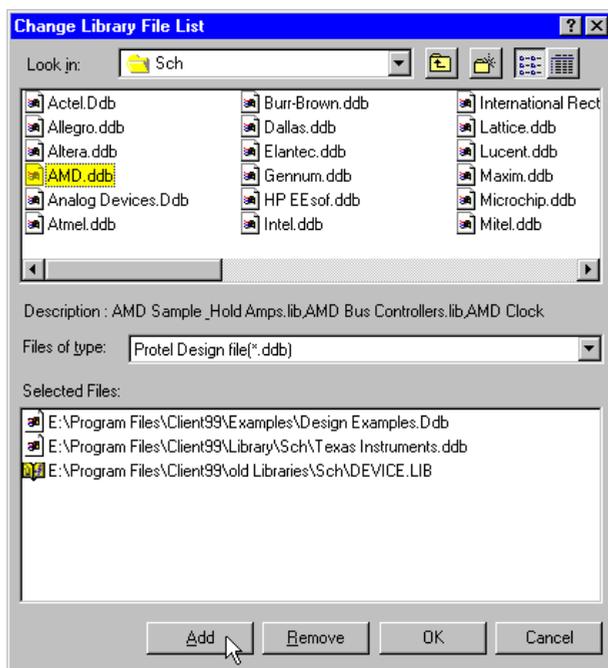


Рис. 3.21. Подключение библиотек к редактору схем.

Подключение и отключение библиотек

Для определения доступа к библиотекам компонентов используется диалоговое окно **Change Library File** (рис. 3.21). В нижней части окна в поле **Selected Files**, показаны все подключенные в настоящее время библиотеки.

Для выбора папки, в которой находятся библиотеки компонентов, используется выпадающий список **Look**, расположенный в верхней части диалогового окна. Система Protel 99 SE хранит библиотечные базы данных в папке **\Program Files\Design Explorer 99 SE\Library\Sch**.

После выбора нужной библиотечной базы данных для подключения ее к системе выполните на ней двойной щелчок левой кнопкой мыши. Для отключения базы данных выполните на ней двойной щелчок левой кнопкой мыши в списке **Selected Files**.

Поиск компонента в библиотеках

Добавить библиотеку легко, если известно, какую библиотеку необходимо использовать, и в какой библиотечной базе данных она находится, но что делать когда неизвестно, в какой библиотеке находится нужный компонент? Редактор принципиальных схем включает мощное средство для поиска компонентов.

Для того чтобы найти нужный компонент в существующих библиотеках необходимо нажать кнопку **Find** или выполнить команду меню **Tools » Find Component** для вызова диалогового окна **Find Schematic Component**.

Советы по поиску нужных компонентов

Поиск при помощи поля **Library Reference**, как правило, работает быстрее.

Так как различные производители используют разные префиксы и суффиксы при именовании своих изделий, для расширенного поиска рекомендуется использовать символ-заменитель "*".

Для увеличения шансов совпадения полезно использовать поле **Description**, причем, здесь лучше задавать сложные условия поиска из нескольких слов, окруженных символами-заменителями "*" и разделенных пробелами (например, *barrel* *shifter*).

Если проведенный поиск не дал результатов, проверьте правильно ли задан путь поиска в поле **Path**. Для того чтобы убедиться, что все сделано правильно, попробуйте найти компонент, заведомо существующий в какой-либо библиотеке.

Для подключения библиотеки с найденным компонентом к системе необходимо использовать кнопку **Add to Library List**.

Размещение секций компонентов на листе принципиальной схемы

И так, требуемые компоненты найдены, библиотеки подключены, и все готово для начала размещения элементов на схеме. Для размещения секции компонента, выберите ее в списке в нижней части рисунка и нажмите кнопку **Place**.

Разместить элемент на схеме можно с помощью команды меню **Place » Part**, или посредством нажатия кнопки **Place Part** на панели инструментов **Wiring Toolbar**, после чего откроется диалоговое окно для выбора компонента библиотеки.

Атрибуты элементов принципиальных схем

Подобно всем объектам в редакторе принципиальных схем, компоненты и секции имеют индивидуальный набор атрибутов. Некоторые из этих атрибутов могут быть определены только в редакторе библиотек, а некоторые — в редакторе принципиальных схем.

Имеется возможность редактировать атрибуты выбранного компонента непосредственно перед размещением его на схеме. Для того необходимо нажать клавишу **TAB**, после чего появится диалоговое окно **Part**. При повторном размещении аналогичных компонентов численная часть позиционного обозначения будет автоматически увеличиваться.

Пока выбранный компонент "приклеен" к указателю мыши, имеется возможность его последовательного поворота на угол 90° посредством нажатия клавиши **SPACEBAR** (пробела), а также зеркального отображения относительно осей X или Y нажатием клавиш **X** и **Y** соответственно.

При размещении секций компонентов без жесткой привязки к номеру секции буквенный суффикс добавляется к позиционному обозначению автоматически, например, U1:A.

Для того чтобы отредактировать атрибуты уже размещенного компонента, необходимо выполнить двойной щелчок левой кнопкой мыши на нем или выполнить команду меню **Edit » Change**. Отредактированным может быть каждый доступный в данный момент атрибут, и изменения могут относиться как к выбранному, так и ко всем компонентам на листе принципиальной схемы. Более подробная информация о глобальном редактировании объектов представлена в подразделе *Глобальное редактирование* раздела *Работа с редактором принципиальных схем*.

Текстовые данные

Любой компонент имеет два набора специальных текстовых полей. Восемь из них (**Text Fields**) вводятся только в редакторе библиотек, а в редакторе схем доступны только для чтения. Другие шестнадцать (**Part Fields**) редактируются на уровне листа принципиальной схемы при размещении компонента.

Имя компонента (Library Reference)

Данный атрибут отражает имя компонента в библиотеке. Длина этого имени не должна превышать 255 символов.

Топологическое посадочное место элемента на плате (Footprint)

Топология посадочного места элемента (**Footprint**) представляет собой реальное изображение корпуса элемента и набора контактных площадок для пайки его выводов на плате, которое будет использоваться в редакторе печатных плат. При генерации списка соединений для последующей разработки платы каждый компонент должен иметь топологию посадочного места, доступную в редакторе библиотек топологий элементов.

Для одних и тех же компонентов может быть заранее задано до четырех различных топологий. Имеется возможность определения топологии при добавлении элемента на схему.

Позиционное обозначение (Designator)

Позиционное обозначение однозначно определяет каждый элемент разрабатываемой схемы. Если перед размещением элемента вы не ввели его специальное (нестандартное) обозначение, то оно будет задано по умолчанию, например, **R?** или **C?**. Используемый по умолчанию префикс позиционного обозначения элемента задается в редакторе библиотек.

Для того, чтобы ввести позиционное обозначение непосредственно перед размещением элемента на схеме, нажмите клавишу **TAB**. При последующем размещении элементов, позиционное обозначение которого редактировалось нажатием клавиши **TAB**, численная часть обозначения будет автоматически увеличиваться, например, **R1**, **R2** и т. д.

Если компонент состоит из нескольких секций, то каждой секции будет автоматически присваиваться суффикс секции, например, **U3A**, **U3B** или **U1:1**, **U1:2** и т. д. Суффикс секции будет автоматически увеличиваться, даже если обозначение было задано непосредственно перед тем размещением секции на схеме.

Для предварительной установки автоматического обозначения выполните команду меню **Tools** » **Annotate**. Более подробная информация по этому поводу приведена в разделе *Передача информации о схеме в редактор печатных плат*.

Тип элемента (Part Type)

Текстовое поле **Part Type** применяется для описания типа элемента (например, 74НС32) или его номинала (например, 220 nF). Длина записи в этом поле не должна превышать 255 символов.

Адрес подчиненного листа (Sheet Path)

Иногда бывает полезно определить элемента схемы не как компоненты, а как символы подчиненных листов. В этом случае проводники, соединенные с выводами таких элементов, могут быть связаны с портами на соответствующих подчиненных листах. Чтобы определить элемент как символ листа, необходимо ввести имя подчиненного листа в поле **Sheet Path** и включить опцию **Descend into sheet parts** в диалоговом окне **Netlist Creation**. Сконфигурированный таким образом элемент, не отражается в списке соединений. Более подробная информация об этом приведена в разделе *Управление многолистовыми и иерархическими проектами*.

Секция (Part)

В этом поле задается номер секции данного компонента. Число 1 соответствует секции А, число 2 — секции В, и т. д. Суффикс секции может быть либо буквой, либо цифрой, что задается опцией **Multi-part Suffix** в диалоговом окне **Preferences**. Следует отметить, что данная установка распространяется на все открытые листы схем.

Скрытые выводы (Hidden Pins)

В редакторе библиотек элементов, выводы компонентов могут быть определены как скрытые (**Hidden**) или видимые (**Visible**). Как правило, скрытыми задаются выводы питания, что позволяет разгрузить принципиальную схему. Скрытые выводы автоматически соединяются с одноименными в процессе создания списка соединений. При включении данного флага скрытые выводы отображаются на экране, после чего они должны быть соединены в ручную.

Если включается отображение скрытых выводов составного компонента, то будут показаны все скрытые выводы всех секций этого компонента.

Скрытые поля (Hidden Fields)

При включении этого флага показываются 16 скрытых по умолчанию полей **Part Fields**, их содержимое отображается на листе принципиальной схемы.

Названия полей (Field Names)

При включении этого флага показываются названия 16 скрытых по умолчанию полей **Part Fields**. Названия полей могут быть отредактированы в редакторе библиотек элементов.

Нередактируемые библиотечные поля (Read Only Library Field)

Каждый библиотечный элемент имеет восемь текстовых полей длиной 255 символов. Библиотечные текстовые поля не могут быть отредактированы в редакторе принципиальных схем, но их содержимое может отображаться на схеме и в перечне элементов (**Bill of Material**).

Текстовые поля (Part Fields)

В дополнение к восьми нередактируемым текстовым полям, существуют шестнадцать текстовых полей длиной до 255 символов каждое. Эти поля можно включать и выключать, редактировать с индивидуальной настройкой типа, размера и цвета шрифтов, а также включать в перечень элементов.

Названия текстовых полей элемента, могут быть определены в редакторе библиотек. Это название может иметь до 255 символов, в окне редактирования будут показаны лишь первые 14 символов.

Ориентация (Orientation)

Ориентация элемента представляет собой угол поворота относительно изначального положения оригинала в библиотеке.

Формат (Mode)

Редактор принципиальных схем поддерживает три формата прорисовки каждого элемента схемы: Normal, DeMorgan и IEEE. Начертание в этих форматах для каждого конкретного элемента задается редакторе библиотек, однако обязательно должен присутствовать только формат Normal. Если при смене формата условное графиче-

ское обозначение какого-либо элемента не изменяется, это говорит о том, что других начертаний у него нет.

Цвета (Colors)

Цвета отображения линий (**Line Color**), выводов (**Pin Color**) и заливки (**Fill Color**) могут быть предварительно определены в редакторе библиотек. При необходимости можно выборочно изменить цвета на принципиальной схеме, для чего надо включить опцию **Local Color** и установить требуемые цвета.

Если условное графическое изображение элемента не имеет заливки (т.е. внутри фигуры виден цвет листа и сетка), то изменения, сделанные в поле **Fill Color** будут игнорироваться.

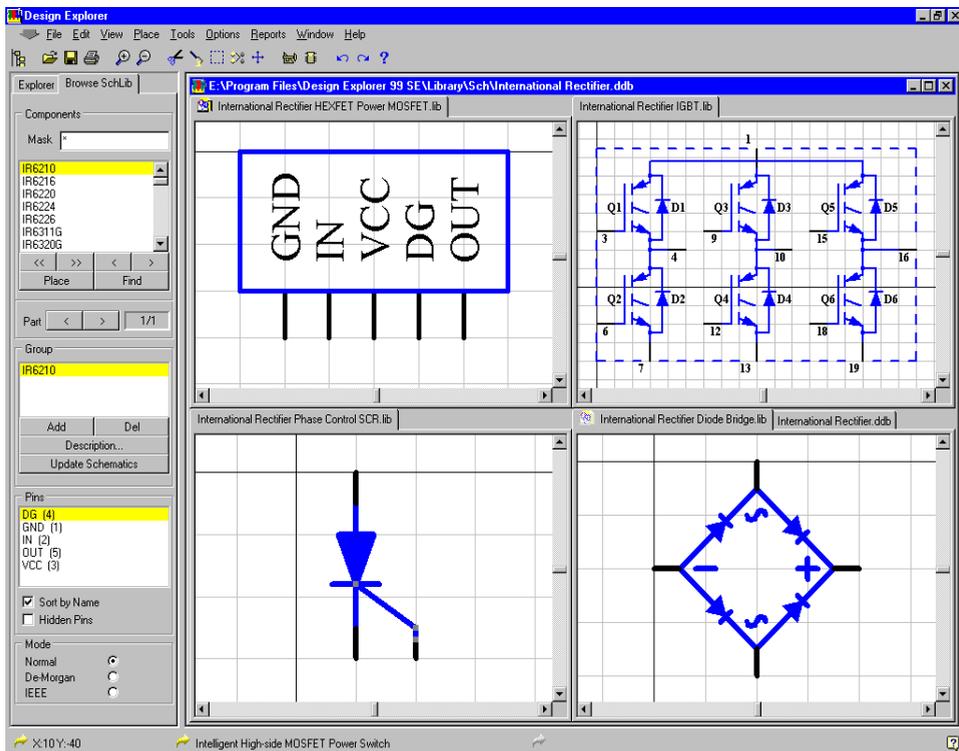


Рис. 3.22. Внешний вид редактора библиотек системы Protel 99 SE.

Редактор библиотек элементов принципиальных схем

Редактор библиотек является вторым редактором документов системы Protel 99 SE, включенным в сервер работы с принципиальными схемами. Первым был редактор

принципиальных схем, используемый для их разработки. Редактор же библиотек используется для создания и изменения компонентов, используемых в этих схемах (рис. 3.22).

В системе Protel 99 SE библиотеки элементов принципиальных схем содержатся в специальных базах данных. Большинство из этих баз данных ориентированы на конкретных производителей, то есть компания разрабатывает единые библиотеки для микросхем того или иного изготовителя. Также существуют специальные базы данных, например, библиотеки графических элементов для проектирования устройств на основе ПЛИС или схемотехнического моделирования.

Работа с библиотеками

В системе Protel 99 SE библиотеки хранятся внутри специальной базы данных, аналогично всем другим документам проекта. В этом справочнике они называются библиотечными базами данных (Library Databases), но на самом деле они представляют собой такие же базы данных, что и обычные документы (схемы, чертежи плат) проекта.

Библиотечные базы данных системы Protel 99 SE находятся в папке `\Program Files\Design Explorer 99 SE\Library\Sch`.

Просмотр существующей библиотеки

Библиотеки в редакторе библиотек открываются аналогично всем остальным документам, открываемым в среде проектирования Design Explorer. Сначала посредством команды меню **File** » **Open** открывается нужная библиотечная база данных, после чего необходимая библиотека элементов выбирается и открывается для редактирования.

Каждая библиотека открывается на новой вкладке в окне проекта.

Создание новой библиотеки

До создания новой библиотеки необходимо открыть проект, в котором нужно сохранить новую библиотеку, и в нем открыть папку, где она будет находиться.

Далее надо щелкнуть правой кнопкой мыши в окне выбранной папки и в появившемся контекстном меню выбрать опцию **New**. В появившемся диалоговом окне **New Document** следует выполнить двойной щелчок левой кнопкой мыши на иконке **SchLib** (рис. 3.23), после чего новая библиотека будет создана.

По умолчанию новым библиотекам в системе Protel 99 SE присваивается имя **SchLib1**. Для того, чтобы переименовать библиотеку, необходимо щелкнуть на ней мышью и нажать клавишу **F2**. После этого подсветится строка, в которой можно задать новое имя библиотеки.

После ввода нового имени нажмите клавишу **ENTER**. Двойным щелчком по иконке откройте только что созданную библиотеку.

Компоненты и секции

Библиотека состоит из набора компонентов. Каждый компонент рассматривается как реальное физическое устройство. Каждый из этих компонентов может включать в себя одну или несколько секций. Например, компонент "резистор" представляет собой всего 1 резистивный элемент, а компонент "матрица резисторов" может содержать восемь резисторов как секции.

Разделение компонентов на элементы всецело лежит на разработчике. Он вправе решать как будет представлять собой компонент. Например, компонент "реле" можно представить как отдельные элементы "контакты" и "катушка". Разъем с четырьмя контактами можно представить как единый элемент, а можно как четыре разные секции. Каждая секция должна быть нарисована на отдельном листе. Для добавления новой секции к компоненту используется команда меню **Tools » New Part**.

Каждая секция компонента может быть представлена условным графическим обозначением в трех различных форматах: Normal, DeMorgan и IEEE. Каждое обозначение рисуется на отдельном листе. Выбор начертания производится при прорисовке принципиальной схемы. По умолчанию устанавливается формат Normal, а значит этот формат является обязательным, а два других — дополнительными.

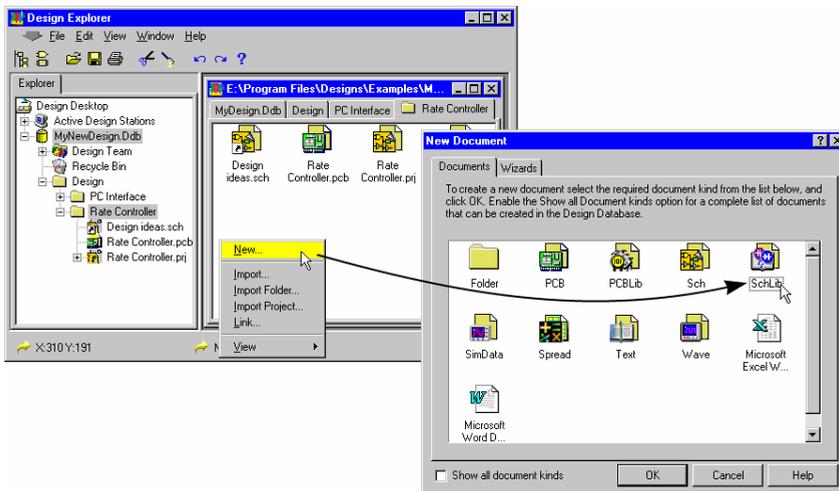


Рис. 3.23. Создание новой библиотеки элементов схем.

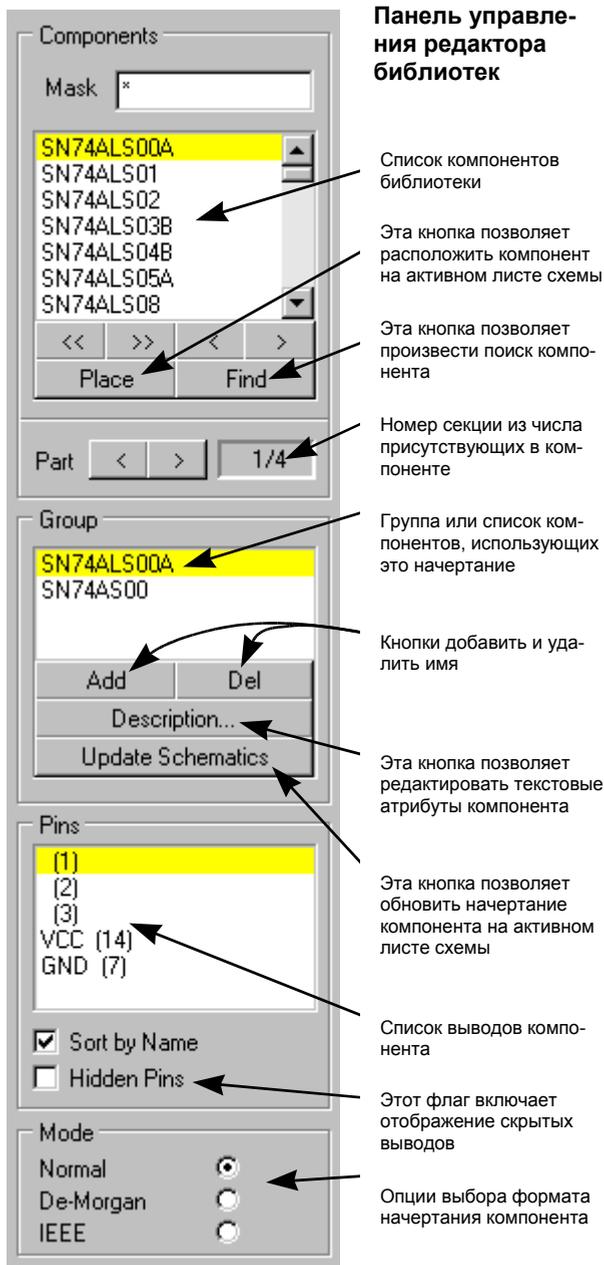


Рис. 3.24. Панель управления редактора библиотек.

Создание и редактирование компонента

В меню **Tools** редактора библиотек элементов принципиальных схем присутствуют все инструменты, необходимые для их создания и редактирования компонентов.

1. Выполните команду меню **Tools » New Component**.
2. Появится чистый лист с именем **Component_1**, на котором будет создаваться компонент. В центре экрана будет присутствовать точка начала координат (**Origin**), обозначенная перекрестьем.
3. Для уменьшения масштаба листа последовательно нажмите клавишу **PageDown** до появления сетки. Если при изменении вида вы случайно потеряли точку начала координат (**Origin**), в нее всегда можно вернуться с помощью команды меню **Edit » Jump » Origin**.
4. Первым шагом в создании нового компонента является прорисовка его тела — главной части условного графического обозначения. Если предполагается компонент прямоугольной формы, нажмите кнопку **Rectangle** на панели инструментов или выполните команду меню **Place » Rectangle**. Иначе прямоугольник можно нарисовать с помощью линий.
5. Щелчком мыши задайте левый верхний угол прямоугольника. Координаты указателя мыши приведены в строке состояния



◆ Если пользователь случайно потерял точку привязки на листе, в нее можно вернуться, нажав горячие клавиши **J**, **O**.

6. Сдвигая указатель мыши вправо и вниз задайте нужные размеры прямоугольника. Как правило, его размер зависит от количества выводов. При рисовании тела компонента рекомендуется пользоваться сеткой, которая позволяет представлять выводы точно в узлах с заданным шагом.
7. Вторым щелчком мыши задайте правый нижний угол прямоугольника.

◆ Изменить размеры нарисованного прямоугольника можно в любое время. Достаточно щелкнуть по нему мышкой, выделить фокусом, а затем потянуть за появившиеся метки-манипуляторы. Для снятия выделения фокусом достаточно щелкнуть мышью вне редактируемого объекта.

8. После размещения тела компонента необходимо разместить его выводы. Для этого на панели инструментов необходимо нажать кнопку **Place Pin**, после чего к указателю мыши неэлектрическим концом "приклеится" вывод, который надо разместить на одной из сторон прямоугольника. Нажатие клавиши **SPACEBAR** позволяет поворачивать размещаемый вывод на угол, кратный 90° . Следует помнить, что электрический конец у вывода только один и он должен быть



расположен вне тела компонента. Неэлектрический конец легко оп-

◆ Уже размещенный вывод можно легко переместить, щелкнув и удерживая левую кнопку мыши. Нажатие клавиши **SPACEBAR** позволяет поворачивать удерживаемый вывод.

ределить по расположению рядом с ним названия вывода.

9. Для задания атрибутов вывода перед его размещением на листе необходимо нажать клавишу **TAB**. Все атрибуты описаны в следующем разделе. Если атрибуты вывода задать до его размещения на листе, то сделанные пользователем установки будут назначены по умолчанию. Это означает, что числовая часть названия вывода будет автоматически увеличиваться.

10. Остальные выводы размещаются аналогично.

Выводы компонентов

Именно выводы придают компоненту электрические свойства. Выводы имеют набор атрибутов, которые устанавливаются в диалоговом окне **Pin**. Атрибуты можно установить как до, так и после размещения вывода на листе. Чтобы установить атрибуты вывода до размещения на листе, необходимо нажать клавишу **TAB** в момент, когда он "приклеен" к указателю мыши. Установка атрибутов уже размещенного вывода проводится двойным щелчком по нему или выбором нужного вывода в списке выводов в браузере проектов.

Имя вывода (Pin Name)

Имя вывода вводить не обязательно, кроме тех случаев, когда вывод планируется сделать скрытым (**hidden**). Скрытые выводы автоматически подключаются к одноименным скрытым выводам и цепям.

Номер вывода (Pin Number)

Номер вывода компонента. Каждый вывод должен иметь свой уникальный номер.

Символ отрицания (Dot symbol)

Атрибут добавляет маленький кружочек (символ отрицания) к выводу.

Символ синхронизации (Clock Symbol)

Атрибут добавляет маленький треугольник (символ вывода синхронизации) к выводу.

Электрический тип вывода (Electrical Type)

Этот атрибут используется при подготовке к проверке правил электрических соединений (**Electrical Rule Check**) в редакторе принципиальных схем. В общем случае этот атрибут устанавливать не обязательно, но если вы намереваетесь использовать проверку правил электрических соединений, его надо установить корректно.

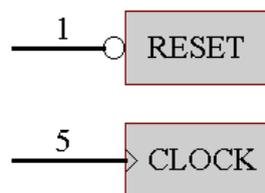


Рис. 3.25. Выводы Dot и Clock.

Скрытый вывод (Hidden)

Включение этой опции делает вывод скрытым. Скрытый вывод автоматически подключается к другим скрытым выводам и цепям с идентичными именами. Как правило, скрытыми определяют выводы питания, чтобы не загромождать принципиальную схему. Включить скрытые выводы на отображение можно с помощью команды меню **View » Show Hidden Pins**.

Показать имя (Show Name)

Включает имя вывода на отображение на принципиальной схеме. В редакторе библиотек элементов имя всегда остается видимым.

Показать номер (Show Number)

Включает номер вывода на отображение на принципиальной схеме. В редакторе библиотек элементов номер всегда остается видимым.

Длина вывода (Pin Length)

Здесь задается длина вывода в сотых долях дюйма.

Описание компонентов

Помимо условного графического обозначения, каждый компонент имеет ряд ассоциированных с ним текстовых полей. Для их просмотра и редактирования пользователь может с помощью команды меню **Tools » Description** вызвать диалоговое окно **Component Text Fields**.

Обозначение, присваиваемое по умолчанию (Default Designator)

Для автоматической простановки позиционного обозначения элемента используется определенный по умолчанию префикс, например, R?, C?, U?.

Топологическое посадочное место (Footprint)

Для задания топологических посадочных мест имеется четыре поля. Если эти поля не были определены в редакторе библиотек, то их значение можно задать после помещения элемента на лист принципиальной схемы. Вышеуказанные четыре поля позволят вам назначить различные топологические посадочные места элемента для различных конструктивных исполнений, например, DIP или SMD.

Адрес подчиненного листа (Sheet Part Filename)

Иногда бывает полезно определить элемента схемы не как компоненты, а как символы подчиненных листов. В этом случае проводники, соединенные с выводами таких элементов, могут быть связаны с портами на соответствующих подчиненных листах. Чтобы определить элемент как символ листа, необходимо ввести имя подчиненного листа в поле **Sheet Part Filename** и включить опцию **Descend into sheet parts** в диалоговом окне **Netlist Creation**. Имя файла элемента схемы можно ввести и после того, как элемент помещен на принципиальную схему. Более подробная информация об этом содержится в главе *Управление многolistовыми и иерархическими проектами*.

Текстовые библиотечные поля

Каждый библиотечный элемент имеет восемь текстовых полей длиной 255 символов. Библиотечные текстовые поля не могут быть отредактированы в редакторе принци-

пиальных схем, но их содержимое может отображаться на схеме и в перечне элементов (Bill of Material).

Текстовые поля элементов (Part Text Fields)

В дополнение к восьми не редактируемым текстовым полям, существуют шестнадцать текстовых полей длиной до 255 символов каждое. Эти поля можно включать и выключать, редактировать с индивидуальной настройкой типа, размера и цвета шрифтов, а также включать в перечень элементов.

Имеется возможность задавать названия текстовых полей элементов, которые могут иметь длину до 255 символов, но в окне редактирования будут показаны лишь первые 14 символов.

Поле описания компонента (Description)

В это поле вводится текстовое описание компонента длиной до 255 символов, которое также можно использовать при поиске нужного элемента в библиотеках.

Что такое группа компонентов?

Некоторые компоненты используют одинаковые корпуса. Они имеют идентичные топологические посадочные места и расположение выводов, но в библиотеках представлены с различными именами. Это могут быть как идентичные устройства от разных производителей, так и компоненты одного семейства, использующие один корпус, но имеющие различные электрические характеристики. Например микросхемы памяти различных версий по скорости: 80 нс и 120 нс. Очень удобно вызывать элемент из библиотеки по его названию (описанию), но рисовать для одинаковых элементов свое графическое представление было бы непозволительной роскошью по причине неэффективности.

Редактор принципиальных схем использует принцип группирования компонентов, чтобы связать множество имен компонентов с единственным описанием, хранящимся в библиотеке. Этот принцип позволяет эффективно использовать и управлять библиотеками. Например, в библиотеке ТТЛ содержится приблизительно 1800 имен компонентов, но они представлены всего лишь 600 графическими представлениями и описаниями.

Копирование компонентов

Компоненты могут быть скопированы из активной библиотеки или из других библиотек с помощью команды меню **Copy » Component**. В диалоговом окне **Destination Library** указывается место, куда будет копироваться компонент. Это может быть та же или другая библиотека. После выбора библиотеки компонент по щелчку кнопки **OK** копируется в указанное место. Заметим, что данное диалоговое окно появляется только когда в редакторе открыто более одной библиотеки.

При копировании компонента внутри одной библиотеки необходимо помнить, что компонент появится с тем же именем. Новый компонент следует переименовать с помощью команды меню **Tool » Rename Component**.

Можно копировать компоненты из одной библиотеки в другую. Для этого в одной библиотеке необходимо на панели управления с помощью мыши выбрать нужные компоненты (допускается использование стандартных для системы Windows комби-

наций клавиш **SHIFT** и **CTRL**), нажать правую кнопку мыши и в появившемся меню выбрать команду **Copy**. Затем следует перейти в другую библиотеку, на панели управления щелкнуть правой кнопкой мыши и выполнить команду **Paste**.

Обновление принципиальных схем

Существует два способа обновления принципиальных схем, после редактирования компонента:

1. Из редактора библиотек: нажмите кнопку **Update Schematics** на панели инструментов редактора библиотек или выполните команду меню **Tools** » **Update Schematics**. После этого представление компонента обновится на всех открытых листах принципиальных схем.
2. В редакторе принципиальных схем: выполните команду меню **Design** » **Update Parts in Cache**, после чего будут обновлены все элементы на всех открытых листах принципиальных схем отличающиеся от элементов из текущего списка библиотек редактора принципиальных схем (а не только библиотек, открытых в редакторе библиотек компонентов).

Отчеты редактора библиотек элементов

Редактор библиотек элементов генерирует три типа отчетов.

Отчет о компоненте

В этом отчете собирается вся доступная информация об активном компоненте. А именно: число элементов, имена групп, подробная информация о выводах для любой секции, входящей в компонент. Файл этого отчета имеет расширение **CMP**.

Отчет о библиотеке

В этом отчете приводится список всех компонентов библиотеки с их описаниями. Файл этого отчета имеет расширение **REP**.

Проверка компонента на наличие ошибок (Component Rule Check)

Для проверки компонента на наличие ошибок пользователь должен установить атрибуты, которые он желает проконтролировать, и нажать кнопку **OK**. Файл такого отчета имеет расширение **ERR**.

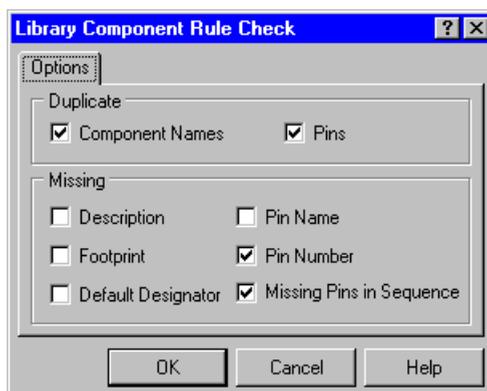


Рис. 3.26. Настройка проверки библиотечного компонента.

Управление многолистовыми и иерархическими проектами

Редактор принципиальных схем системы Protel 99 SE позволяет работать как с одно- и многостраничными схемами, так и с проектами, имеющими сложную иерархическую структуру (рис. 3.27).

Максимальное количество схем в проекте ограничивается лишь объемом жесткого диска.

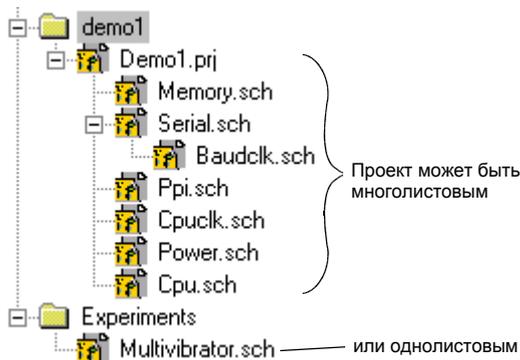


Рис. 3.27. Различные типы проектов Protel 99 SE.

Общие положения

Каждый лист принципиальной схемы сохраняется как отдельный документ. Любой лист схемы можно открыть и редактировать независимо от других простым щелчком по соответствующей иконке в окне просмотра дерева проекта.

Проект может состоять из единственного листа принципиальной схемы или из нескольких листов с многочисленными связями между ними.

Многолистовые проекты предназначены для описания сложных систем, схемы которых не могут быть размещены на одном листе. Даже если разрабатываемая система не слишком сложная ее схему выгодно разбить на несколько листов. Например, устройство может состоять из нескольких модулей, распределение которых по отдельным схемам позволит нескольким инженерам-разработчикам одновременно работать над проектом. Другой причиной, по которой следует разбивать проект на несколько листов, является удобство получения распечатки схемы на малоформатных лазерных принтерах.

Если два или более файла принципиальных схем каким-то образом объединены или связаны между собой, то они представляют собой один многолистовой проект. Существует несколько способов организации многолистовых проектов, причем, выбор конкретного способа разбиения зависит от типа, размера и структуры проекта.

Управление многолистовыми проектами

Управление проектами — это процесс задания и обработки связей между отдельными листами, входящими в состав проекта. Эти межлистовые соединения определяют единую связность, а также ряд других преимуществ при управлении проектом, например, способствуют перемещению по проекту, просмотру и доступу к любому листу проекта, а также позволяют генерировать сквозной список соединений и провести проверку правил электрических соединений (Electrical Rule Check).

Межлистовые связи позволяют открывать и сохранять весь проект посредством выполнения всего одной операции, а также выполнять глобальное редактирование и поиск текста во всех листах.

- ◆ Многие функции редактора принципиальных схем, такие как обновление позиционных обозначений и распечатывание, применимы только к открытым листам. При выполнении этих операций автоматически будет открываться весь проект, включая главный лист и все листы, связанные с ним.

Структура многолистовых принципиальных схем

Главным преимуществом многолистовой среды редактора схем является его интуитивность, которая значительно облегчает работу со сложными иерархическими проектами. Все многолистовые проекты включают в себя специальный файл, называемый главным листом (**master sheet**). Главный лист находится на вершине иерархии проекта и является его первым листом. Термин иерархия указывает на отношения между главным и подчиненными листами, составляющими проект.

Особый класс иерархии, называемой комплексной, позволяет иметь несколько ссылок на один подчиненный лист принципиальной схемы. Эта иерархическая структура может принимать различные формы, в зависимости от метода объединения листов. Детальная информация по использованию комплексной иерархии будет описана в подразделе *Различные методы построения многолиствого проекта*.

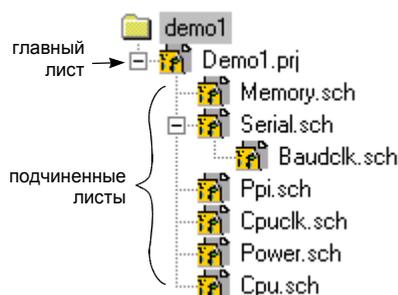


Рис. 3.28. Пример проекта со сложной иерархической структурой.

- ◆ Иерархическое устройство проекта поддерживает истинно модульный подход, позволяя пользователю работать с функциональными блоками. Функциональные блоки имеют пространственные связи на листах принципиальных схем, которые поддерживают обе методологии конструирования как сверху вниз, так и снизу вверх.

Главные и подчиненные листы

Отношения между листами в многолистовом проекте формируются с помощью специального символа, называемого символом листа. Этот символ на старшем в иерархии листе обозначает принципиальную схему подчиненного листа.

Кроме графических атрибутов (цвет, размер, расположение и т. д.) символ листа имеет два дополнительных поля: **Name** и **Filename**. Поле Name является текстовой меткой и используется только для ссылок. Поле Filename указывает на файл принципиальной схемы, представленный символом листа на главном листе. Это поле создает связь между главным и подчиненным листом, определяя тем самым структуру проекта (рис. 3.29).

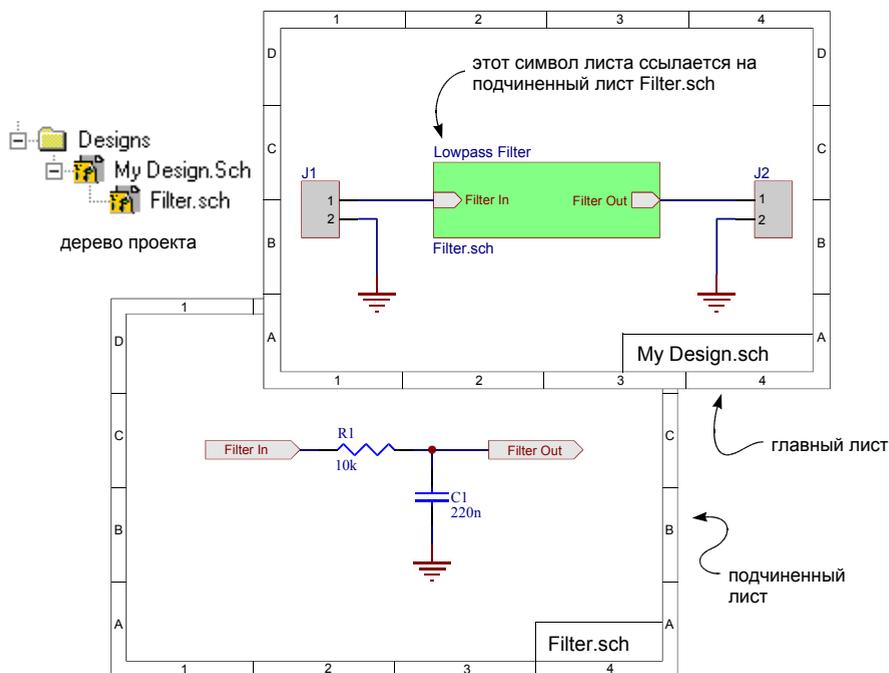


Рис. 3.29. Пример символа листа, ссылающегося на подчиненную схему.

Добавление подчиненного листа в проект

Для того, чтобы добавить открытый лист принципиальной схемы в проект, необходимо поместить символ листа на главный лист. Затем двойным щелчком мыши по схемному символу войти в режим редактирования и ввести имя файла подчиненного листа в поле **Filename** (например **SUBSHEET1.SCH**). В окне просмотра дерева проекта автоматически отобразится новый, только что подключенный, лист.

Как создается связанность в многолистовом проекте?

Редактор принципиальных схем позволяет задавать связи между различными листами в многолистовых проектах различными способами. Соединения типа цепь-цепь (**net-to-net**) задаются идентификаторами цепей (**Net Identifiers**), описание которых приведено ниже.

При создании списка цепей разработчик должен указать редактору схем, как соединять идентификаторы цепей, то есть определить область действия идентификатора цепей (**Net Identifier Scope**). По сути, существует два пути соединения идентификаторов цепей: либо по горизонтали непосредственно между подчиненными листами, либо по от подчиненного к вышестоящему листу. При горизонтальной связанности соединения создаются между метками цепей (**net labels**) или между портами. При вертикальной — между портами и входами листов (**sheet entry**).

По началу управление иерархическими многолистовыми принципиальными схемами может показаться сложным. Однако, однажды поняв несколько основных принципов, пользователю в дальнейшем будет относительно легко пользоваться всей мощностью иерархии при организации сложных проектов.

Идентификаторы цепей (Net Identifiers)

На рисунке 3.30 показаны четыре из пяти возможных идентификаторов цепей. Пятый тип — скрытые выводы компонентов.

Идентификаторы цепей являются связующей субстанцией между отдельными цепями на принципиальных схемах. Идентификаторы цепей помещаются, для того чтобы соединить объекты, принадлежащие к одной и той же цепи либо внутри одного листа схемы, либо на двух и более листах схем в иерархическом проекте. Соединения могут быть физическими (когда один объект непосредственно подсоединен проводником к другому объекту) или логическими (когда идентификаторы цепей показывают связь с другой цепью, имеющей то же имя).

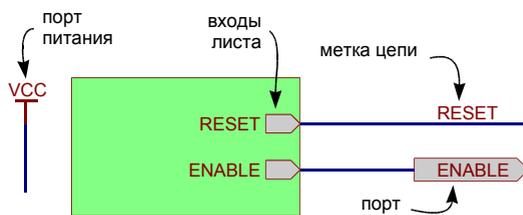


Рис. 3.30. Связность задается идентификаторами цепей различных типов.

Идентификаторами цепей являются:

Метки цепей (Net Label)

Метка цепи используется для уникального определения цепи. Цепь с заданной меткой соединится к другой цепи с тем же именем на том же листе, а так же может подсоединиться к цепям с тем же именем на других листах. Метки цепей можно подсоединять к отдельным проводам, выводам элементам и шинам.

Порты (Port)

В зависимости от метода связанности, порт может быть соединен по горизонтали с другим портом с тем же именем или по вертикали со входом листа (**sheet entry**) с тем же именем.

Входы листа (Sheet Entry)

Входы листа используются для подключения к порту с тем же именем на подчиненном листе при вертикальной связанности.

Порты питания (Power Port)

Все порты питания с одинаковыми названиями соединяются друг с другом во всем проекте.

Скрытые выводы (Hidden Pin)

Скрытые выводы ведут себя как порты питания, глобально, по всему проекту присоединяясь к цепям с аналогичными названиями.

Инверсные метки цепей

Существует два метода создания инверсных (то есть с небольшой горизонтальной чертой над названием) меток цепей, входов листа и портов. Первый заключается в добавлении в название метки цепи символов \ вокруг всех букв (например, "\inlet"). Иначе можно включить опцию **Single ' \ Negation** в диалоговом окне **Preferences**, после чего инверсия будет применена ко всем символам, расположенным после символа \ (например, "\net").

Область действия идентификаторов цепей

Связанность, получающаяся при создании списка соединений в многолистовом проекте, зависит от области видимости идентификаторов цепей. Область видимости определяется тем, как разработчик желает соединить идентификаторы цепей: локально — в пределах листа; глобально — во всех листах; вертикально — между портом и соответствующим входом в лист.

Например, если метки цепей являются локальными, то тогда цепь с названием **Clock1** на одном листе не будет соединяться с цепью таким же названием **Clock1** на другом листе. В этом случае область видимости меток цепей является локальной.

Кроме того, метки цепи могут быть определены и как глобальные идентификаторы. Если это так, то все цепи с меткой **Clock1** (на всех листах проекта) будут считаться частью одной цепи. Порты питания и скрытые выводы всегда определены как глобальные идентификаторы.

Область видимости идентификаторов цепей определяется при генерации списка цепей (**Design » Create Netlist**) или при запуске проверки электрических цепей на наличие ошибок (**Tool » ERC**).

Область видимости идентификаторов цепей должна быть определена на начальной стадии разработки проекта. Ниже описаны различные методы построения проекта.

Различные методы построения многолиствого проекта

Взаимоотношения между листами проекта и идентификаторами цепей наилучшим образом иллюстрируются на примерах пяти возможных моделей организации схемы проекта.

Необходимо помнить, что все проекты с многолистовыми принципиальными схемами имеют иерархическую организацию. Даже если количество уровней иерархии ограничено двумя, то главный лист будет содержать символы листа всех подчинённых листов. Модели 1 и 2, показанные ниже, отображают так называемую плоскую (flat) модель, поддерживаемую программами OrCAD (Модель 1) и Protel DOS Schematic (Модель 2).

Модель 1 — межлистовые соединения определяются с помощью глобальных портов

Модель 1 называется "плоской" из-за того, что все подчиненные листы находятся на одном уровне в иерархии проекта. Верхний лист содержит символы листов для всех подчинённых листов, но не содержит никаких проводников или цепей (рис. 3.31).

В этой модели все межлистовые соединения выполнены глобально с помощью портов, то есть порты с одинаковыми именами соединены между собой по всему проекту. Заметим, что имена цепей в двух подчинённых листах локальны. Это значит, что цепи соединяются только внутри своего листа, а не между листами проекта.

Эта модель рассматривает проект, как если бы он располагался на одном большом листе, разрезанном на отдельные страницы. Такой подход хорошо работает только при разработке небольших проектов, а управление большими проектами здесь затруднительно, так как сложно производить трассировку цепей от одного листа к другому. Для упрощения просмотра цепей в таких проектах рекомендуется включать на листы перекрестные ссылки на соответствующие порты.

Перекрестные ссылки между портами разных листов

Межлистовые связи в плоских проектах могут задаваться двумя методами, представленными в меню Reports редактора принципиальных схем:

Add Port References (Flat) – эта опция добавляет к каждому порту текстовые строки, содержащие имена листов и кодированное обозначение позиций портов связанных портов на них.

Add Port References (Hierarchical) – эта опция добавляет к каждому порту текстовые строки, содержащие имена листов и кодированное обозначение позиций связанных входов листов.

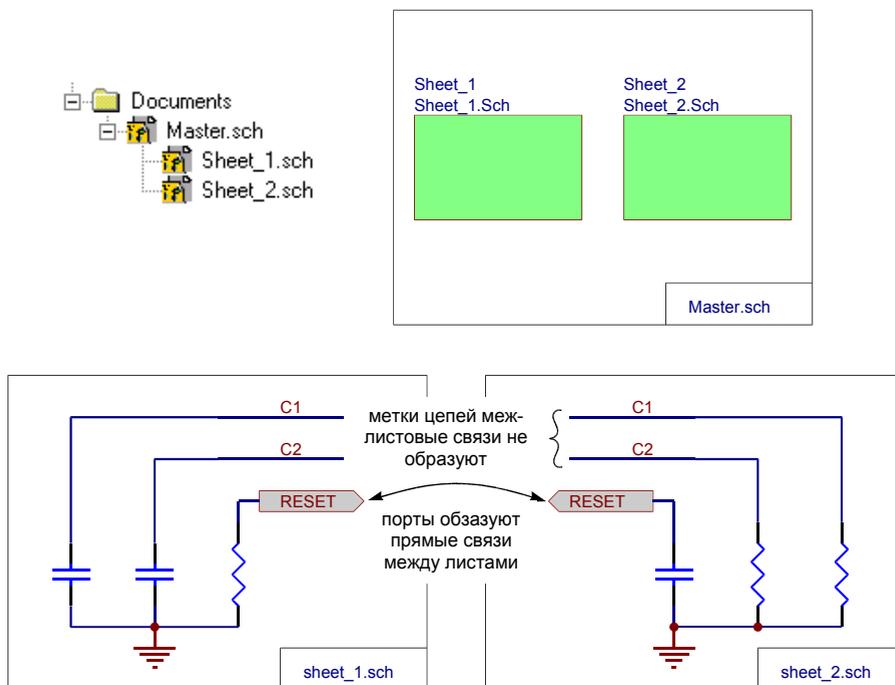


Рис. 3.31. Данная модель описывает "плоский" проект. На главном листе связи между подчиненными листами отсутствуют. Для этой цели могут быть использованы только порты, а все связи получаются горизонтальными.

Ссылка на порт может быть удалена в любой момент работы с проектом с помощью команды меню **Reports » Remove Port References**. Текст ссылок на порты получается автоматически и не может быть изменен и сохранен в проекте. Местоположение текста определяется расположением порта и способом подключения к нему проводника.

Установка области действия идентификаторов цепей

Для этой модели при выполнении проверки правил электрических соединений (ERC) или же при использовании синхронизатора для обновления информации в редакторе печатных плат необходимо область действия идентификаторов цепи установить в режиме **Ports Only Global** (глобальные только порты).

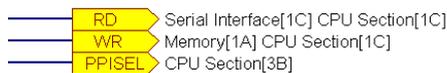


Рис. 3.32. Использование ссылок на порты

Модель 2 — межлистовые соединения определяются с помощью глобальных портов и меток цепей

Модель 2 также называется "плоской" из-за того, что все подчиненные листы находятся на одном уровне в иерархии проекта. Верхний лист содержит символы листа для всех подчинённых листов, но не содержит никаких проводников или цепей. В модели 2 межлистовые соединения создаются как портами, так и метками цепей (рис. 3.33).

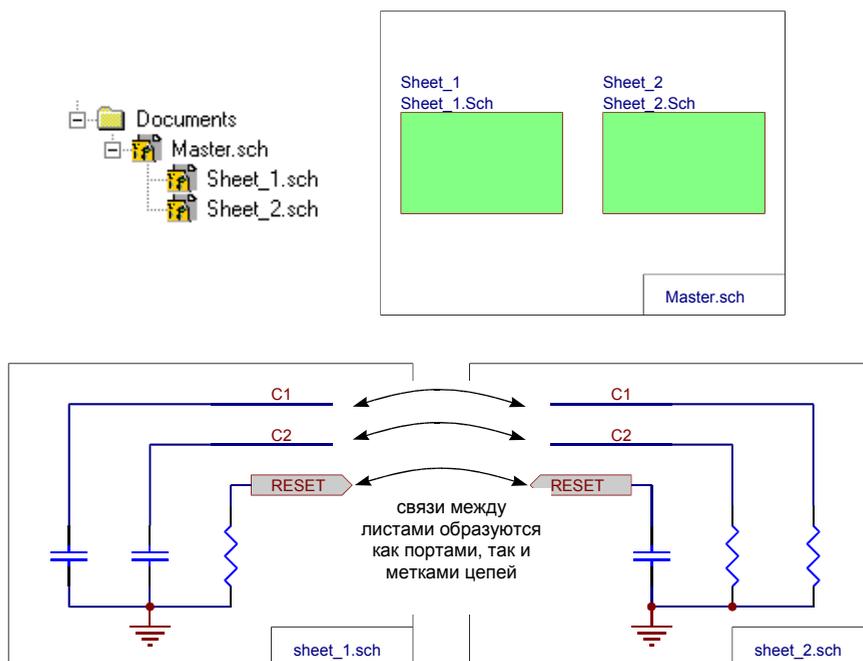


Рис. 3.33. Данная модель описывает "плоский" проект. На главном листе связи между подчиненными листами отсутствуют. Для этой цели используются как порты, так и метки цепей, а все связи получают горизонтальными. Вся информация о схеме представлена в подчиненных листах.

Установка области действия идентификаторов цепей

Для этой модели при выполнении проверки правил электрических соединений (ERC) или же при использовании синхронизатора для обновления информации в редакторе печатных плат необходимо область действия идентификаторов цепи установить в режиме **Net Labels and Ports Global** (глобальные порты и метки цепей).

Модель 3 — простая иерархия

Модель 3 относится к простой иерархии. Эта модель позволяет строить многоуровневые или модульные проекты, где иерархия может быть представлена древовидной

структурой. В этой модели символ листа обозначает подчиненный лист, который спускается из вышестоящего листа (рис. 3.34).

Все межлистовые соединения здесь являются вертикальными, то есть входы листа в каждом символе листа соединяются с аналогичными же портами на соответствующих подчинённых листах.

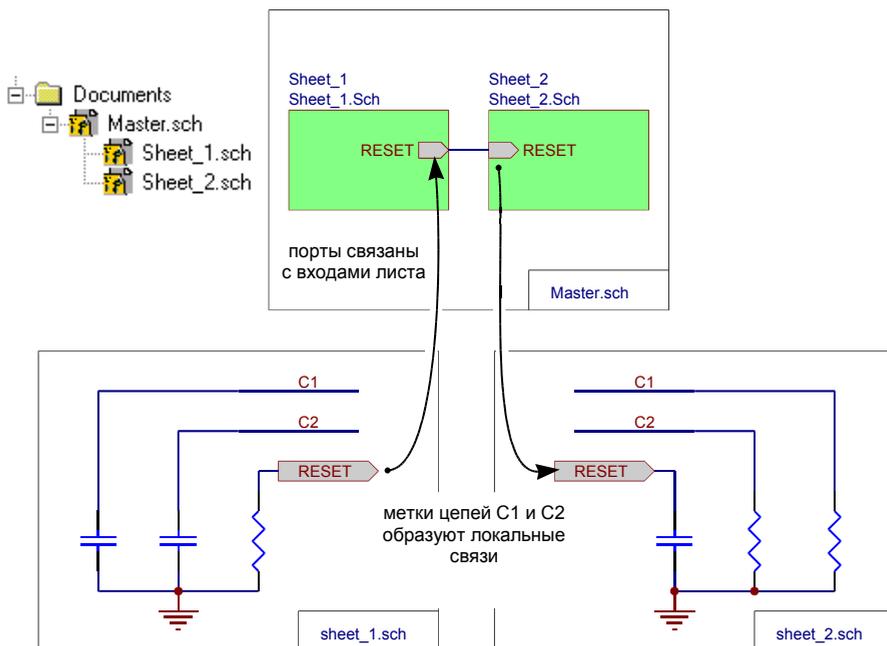


Рис. 3.34. Данная модель описывает проект с простой иерархией, когда каждый лист схемы представлен в проекте только один раз. Связи между листами реализуются как связи между портами и соответствующими входами листа, а также между входами листов на главном листе. Все метки цепей локальные

- ◆ Модель 3 является истинно иерархической, потому что межлистовые соединения сами собой следуют иерархии листов, а проект может быть в глубину настолько многоуровневым, насколько пожелает разработчик. Эта модель наилучшим образом подходит для построения многolistовых проектов.

Установка области действия идентификаторов цепей

Для этой модели при выполнении проверки правил электрических соединений (ERC) или же при использовании синхронизатора для обновления информации в редакторе печатных плат необходимо область действия идентификаторов цепи установить в режиме **Sheet Symbols/Port Connections** (соединения символов листов с портами).

Модель 4 — сложная иерархия

Модель 4 относится к сложной иерархии. В этой модели проект может содержать несколько символов листа с одинаковыми названиями (подчиненных листов). Эти символы могут располагаться либо на одном листе, либо на разных листах многолиствого проекта. Каждому подчиненному символу листа должен быть назначен свой ярлык.

Создание проекта со сложной иерархией можно проиллюстрировать следующим примером (рис. 3.35). Символ листа **Channel_A** ссылается на лист схемы **Channel_A.sch**. Скопируйте этот символ листа на этот же лист схемы и измените ссылку в поле **Filename** на **Channel B.sch**.

Откройте папку, содержащую подчиненный лист **Channel_A.sch**, скопируйте его в буфер обмена и вставьте как ярлык в эту же папку. Измените имя ярлыка на **Channel B.sch** и нажмите клавишу **F5**, чтобы обновить иерархию проекта. Легко видеть, что система поняла переименованный ярлык как новый подчиненный лист.

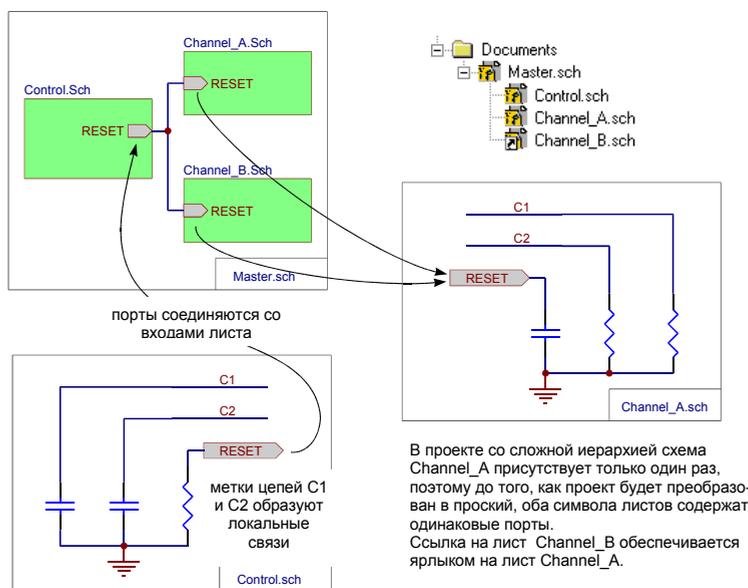


Рис. 3.35. Сложная иерархия используется, когда какой-либо из подчиненных листов используется в проекте более одного раза. Каждый вход листа соединяется с одной именной цепью на подчиненных листах. Все метки цепей локальные.

Эта модель соответствует проектам с высокой степенью модульности. В качестве примера можно привести стереоусилитель, в котором левый и правый каналы имеют одинаковые принципиальные схемы.

Преобразование сложной иерархии в простую

Сложная иерархия используется, главным образом на этапе построения принципиальной схемы. Когда разработчик подошел к этапу создания списка соединений, он должен сделать проект "плоским", то есть преобразовать сложную иерархию в простую. Это можно сделать с помощью команды меню **Tools » Complex to Simple**. Каждый подчиненный лист, использующийся в проекте более одного раза, будет скопирован и переименован. Для присвоения каждому элементу схемы уникального позиционного обозначения необходимо выполнить операцию обновления обозначений.

Установка области действия идентификаторов цепей

Для модели 4 после преобразования проекта из сложной иерархии в простую при выполнении проверки правил электрических соединений (ERC) или же при использовании синхронизатора для обновления информации в редакторе печатных плат область действия идентификаторов цепи также необходимо установить в режиме **Sheet Symbols/Port Connections** (соединения символов листов с портами).

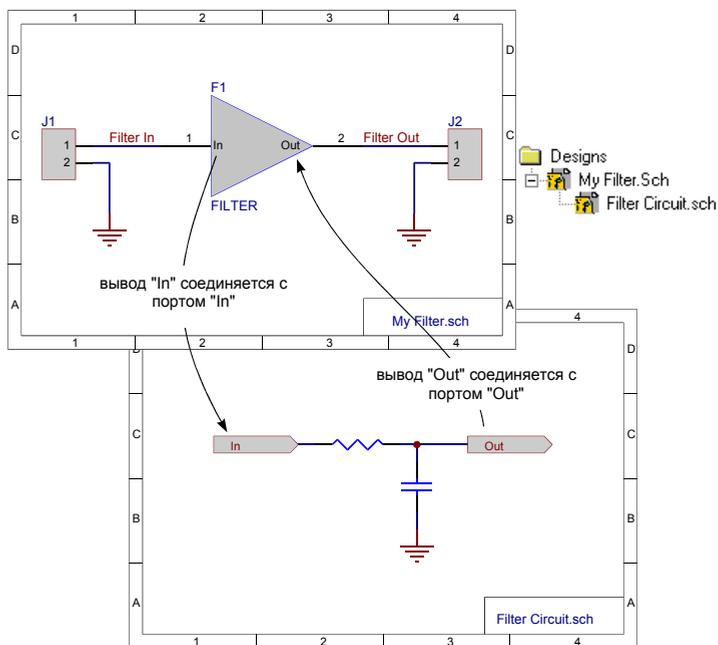


Рис. 3.36. Модель 5 позволяет использовать элемент схемы как символ листа.

Модель 5 – использование символов элементов (Sheet Parts) для создания иерархии

Это особая модель иерархии, в которой компонент ведёт себя как символ листа, указывающий на содержащиеся в нём цепи. В реальной жизни необходимость в использовании такой модели иерархии встречается довольно часто. Предположим, что в пользовательском проекте есть небольшая подчиненная схема, разработку и моделирование которой полезно провести в рамках единого проекта, или же в проекте используется программируемая логическая микросхема, которая должна быть включена с учетом внутренней схемотехники, или разработчик хочет представить схему подчиненного листа не прямоугольником, а любым другим символом (рис. 3.36).

При создании списка соединений у разработчика есть выбор либо включать цепи подчиненного листа в список соединений, при этом сами компоненты в список не войдут, либо считать эти элементы за компоненты. В последнем случае компоненты будут включаться в список цепей, а входящие в них схемы — нет.

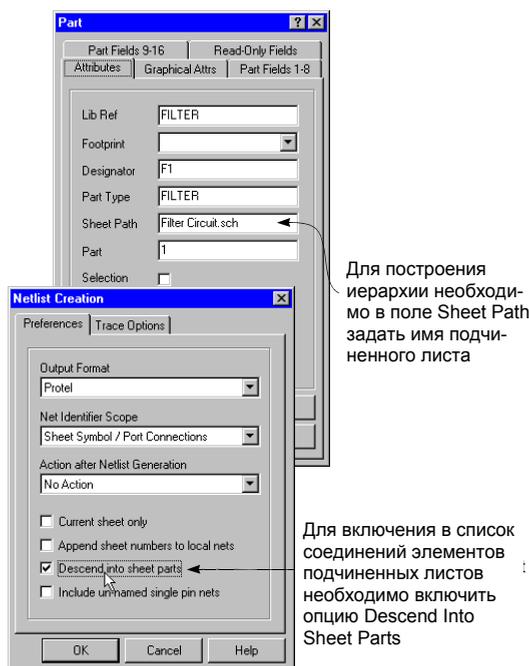


Рис. 3.37. Настройка генерации списка соединений при использовании модели 5.

Для того чтобы воспользоваться этой моделью, необходимо выполнить следующие шаги:

1. На вышестоящем листе в поле **Sheet Path** диалогового окна **Part** ввести имя подчиненного листа.

2. Включить опцию **Descend into Sheet Parts** в диалоговом окне **Netlist Creation**. При использовании этой опции все листы должны располагаться в одной папке (рис. 3.37).

Выводы компонентов будут вести себя как входы листа и соединяться с одноимёнными портами на листах принципиальных схем, указанных в поле **Sheet Path**.

Указать имя файла подчиненного листа можно также и в редакторе библиотек при создании элемента, выполнив команду **Tools » Description** для редактирования имени элемента листа (**Sheet Part Filename**).

Установка области действия идентификаторов цепей

Для модели 5 при выполнении проверки правил электрических соединений (ERC) и создании списка соединений область действия идентификаторов цепей также необходимо установить в режиме **Sheet Symbols/Port Connections** (соединения символов листов с портами). Перед этим необходимо выполнить преобразование проекта из сложной иерархии в простую с включенной опцией **Descend into Sheet Parts**.

Резюме

Модели 3, 4 и 5 описывают различные способы организации сложных проектов. Символы листа в иерархических проектах могут представлять собой функциональные блоки, у которых входы листа служат соединительными звеньями, связывающими цепи на вышестоящем листе с цепями на подчиненном листе.

Эту иерархическую структуру можно объяснить иначе, если представить первый лист в качестве главного, а лист изображаемый символом листа как подчиненный. Пользуясь терминологией иерархических проектов, можно сказать, что подчиненный лист исходит из вышестоящего. При дальнейшем расширении этой модели подчиненный лист может иметь свои собственные подчиненные листы — дополнительные листы, которые спускаются по ветвям этой вертикальной структуры все ниже и ниже.

Как показано выше иерархия может быть как простой, в которой каждый лист является уникальным как в Модели 3, так и сложной, в которой одни и те же подчиненные листы встречаются в проекте неоднократно — модульный подход описанный Моделью 4.

Работа с иерархическим проектом

Редактор принципиальных схем имеет широкий набор инструментов, облегчающих работу с многолистовыми иерархическими проектами. В него входят средства просмотра проекта, создания подчиненных листов и символов листов, а также для преобразования сложной иерархии в простую.

Просмотр проекта

Для того, чтобы сделать иерархический проект удобным, необходимо иметь набор средств для маршрутизации и просмотра иногда весьма запутанных взаимосвязей между множеством объединённых в одном проекте листов. В системе Protel 99 SE

имеются два инструмента, предназначенных для этой цели: окно просмотра дерева проекта **Design Explorer Navigation Panel** и кнопка **Up/Down Hierarchy**.

Использование окна просмотра дерева проекта

Открытие и просмотр листа принципиальной схемы выполняется одним щелчком левой кнопки мыши на соответствующем значке в окне просмотра дерева проекта (рис. 3.38).

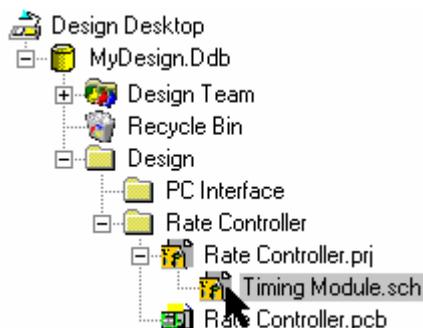


Рис. 3.38. Навигация в сложной иерархии с помощью дерева проекта.

Использование кнопки Up/Down Hierarchy

Для перемещения вверх и вниз по иерархическому дереву служит кнопка **Up/Down Hierarchy**, находящаяся на главной панели инструментов.



При щелчке по этой кнопке система предложит пользователю выбрать порт, вход листа, символ листа или же элемент, используемый в качестве символа листа. Если выполнить щелчок левой кнопкой мыши по входу листа, то на экране появится соответствующий порт на подчиненном листе. Если щелкнуть по символу листа или по элементу, используемому в качестве символа листа, откроется весь подчиненный лист.

Для перемещения вверх по иерархическому дереву проекта необходимо выполнить щелчок левой кнопкой мыши по порту, соединенному с соответствующим входом листа на вышестоящем листе.

Создание символов листов и подчиненных листов — самый простой способ

Редактор принципиальных схем имеет возможность автоматизации процесса построения проекта.

Построение проекта сверху вниз

При построении проекта сверху вниз пользователь начинает проект с верхнего листа и располагает в нем функциональные блоки, используя символы листа. Для автоматического создания подчиненных листов можно воспользоваться командой меню **Design » Create Sheet From Symbol**.

При выборе этой команды меню программа предложит разработчику выбрать символ листа, после щелчка, по которому редактор схем откроет новый лист принципиальной схемы с соответствующим именем файла. Новый подчиненный лист будет содержать все порты, соответствующие входам листа на вышестоящей схеме.

Построение проекта снизу вверх

При построении проекта снизу вверх пользователь уже имеет подчиненный лист и для его представления на верхнем листе необходимо создать символ листа. Это можно сделать с помощью команды меню **Design » Create Symbol From Sheet** (вышестоящий лист при этом должен быть активным). На экране появится диалоговое окно **Choose Document to Place**.

После этого разработчику потребуется указать лист принципиальной схемы, а также указать необходимость изменения направления связей (**Reverse Input/Output Directions**). После этого появится "приклеенный" к указателю мыши символ листа, который следует расположить на схеме. Символ листа будет иметь соответствующее имя файла для связи с подчиненным листом, а также входы листа для всех имеющихся на подчиненном листе портов.

При ответе на вопрос об изменении направления связей (**Reverse Input/Output Directions**) пользователь должен учитывать следующее. Каждый из портов подчиненного листа может быть входным или выходным. По умолчанию тип порта устанавливается как выходной (**output**). При положительном ответе на вопрос об изменении направления вывода (**Yes**) вход листа, соответствующий этому порту, получит тип входа (**input**), то есть направление связи изменится (срез внутрь). При отрицательном ответе на вопрос (**No**) вход листа, соответствующий этому порту, получит тип выхода (**output**) (срез наружу).

Верификация принципиальных схем

Верификация — это важный этап разработки проекта, гарантирующий правильность введенной принципиальной схемы и корректное создание списка соединений. Редактор принципиальных схем включает модуль, выполняющий проверку правильности электрических соединений (Electrical Rule Checker, ERC) на предмет выявления различных ошибок и несоответствий, например, несоединенных входов элементов или замыканий между разными цепями.

Для настройки и запуска процедуры проверки электрических соединений используется команда меню **Tools » ERC**, которая вызывает диалоговое окно **Setup Electrical Rule Check**. В результате проверки генерируется текстовый файл отчета, содержащий предупреждения об ошибках и логических несоответствиях или для активного листа,

или для всего проекта, а также на схеме расставляются специальные метки ошибок, помогающие пользователю найти место ошибки в проекте.

Определение точек, которые не будут восприниматься как ошибки

У пользователя может возникнуть необходимость определить несколько точек на схеме, которые при проверке не должны помечаться как ошибочные. Для этого с помощью команды меню **Place » Directive » No ERC** в этих точках схемы ставятся специальные объекты — директивы, отменяющие проверку правил электрических соединений. Можно запретить вывод на печать этих директив, для чего нужно выключить опцию **Include on Printout** в диалоговом окне **Printer Setup**.

Опции программы верификации

Механизм проверки правил электрических соединений обнаруживает большое количество электрических ошибок различных типов, например, замыкания цепей и наличие неиспользуемых входов логических элементов.

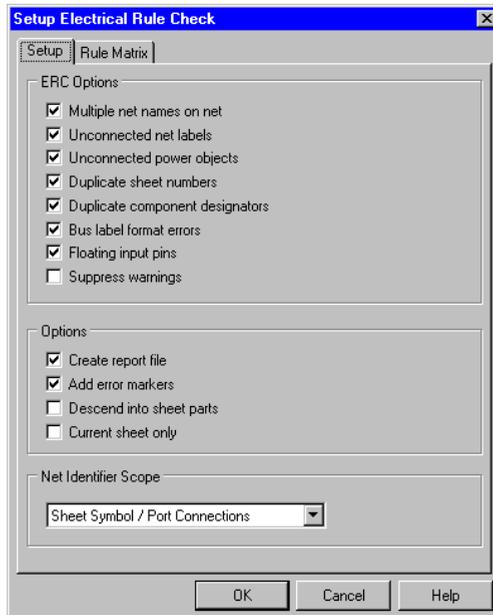


Рис. 3.39. Настройка параметров проверки правил электрических соединений.

Настройка программы проверки правил электрических соединений

Команда меню **Tools** » **ERC** вызывает диалоговое окно **Setup Electrical Rule Check**, в котором определяются различные опции, диапазон действия и параметры процедуры проверки правил электрических соединений (рис. 3.39). Опции включают:

Несколько разных имен одной и той же цепи (Multiple Net Names On Net)

В отчет помещаются физические цепи, обозначенные несколькими идентификаторами цепей с разными именами.

Не подсоединенные метки цепи (Unconnected Net Labels)

В отчет помещаются метки цепи, которые физически не соединены ни с одним электрическим объектом на листе.

Не подсоединенные объекты питания (Unconnected Power Objects)

В отчет помещаются объекты питания, которые физически не соединены ни с одним электрическим объектом.

Одинаковые номера листов (Duplicate Sheet Numbers)

В отчет помещаются листы, которым присвоены одинаковые номера (соответствующее диалоговое окно вызывается командой меню **Design** » **Option**, вкладка **Organization**).

Одинаковые позиционные обозначения элементов (Duplicate Component Designators)

В отчет помещаются элементы с одинаковыми позиционными обозначениями. Подобные ошибки могут появиться в случае, если не использовался загрузчик процедуры **Annotate** или загрузчик, вызываемый с помощью команды меню **Tools** » **Complex to Simple**, который упрощает (делает "плоским") сложный иерархический проект (проект с дублированными листами).

Ошибки формата обозначений шин (Bus Label Format Errors)

В отчет помещаются соединенные с шиной метки цепи, которые неправильным образом отформатированы для отображения набора сигналов в данной шине. Прикрепление меток цепи к шине осуществляется для определения логической связанности для данной шины. Как правило, такие метки цепи включают все сигналы шины. Например, **HA[0..19]** представляет имена сигналов **HA0**, **HA1**, **HA2** и т.д. до **HA19**.

Не подключенные входы (Floating Input Pins)

В отчет помещаются не подключенные выводы, тип (**Electrical Type**) которых задан как **Input**.

Подавление предупреждений (Suppress Warning)

При включении этого флага генерация отчета и расстановка маркеров ошибок производится только при условии бесспорной ошибки, тогда как при снятии этого флага происходит поиск мест в схеме, где по мнению программы проверки возможна ошибка, хотя явного несоответствия правилам электрических соединений нет. Такие вероятные ошибки помечаются как предупреждения (**Warning**) (смотри ниже **Connected Pin / Sheet Entry / Port Rule Matrix**). При включении этой опции процесс проверки проходит быстрее.

Другие опции

Создавать файл отчета (Create Report File)

При установке этого флага генерируется текстовый отчет, в котором сведена вся информация, полученная из программы проверки правил электрических соединений.

Добавлять маркеры ошибок (Add Error Markers)

При установке этого флага на листе схемы размещаются специальные маркеры в местах беспорных и возможных ошибок, отраженных в отчете. В редакторе схем имеются специальные средства, позволяющие легко перемещаться от маркера к маркеру.

Разрешить использование компонентов в качестве символов листа (Descend Into Sheet Parts)

Установка этого флага заставляет трактовать некоторые компоненты на схемах как иерархические символы листов. Для того, чтобы система восприняла элемент схемы как символ листа, необходимо в поле **Sheet Part** указать имя подчиненного листа. Для получения дополнительной информации обратитесь к главе *Управление многолистовыми и иерархическими проектами*.

Настройка области действия идентификаторов цепей

Область видимости идентификаторов цепей определяет метод связанности, используемый в многолистовом проекте. Выбираемая область видимости должна быть такой же, как и при создании списка цепей.

Полное освещение этого вопроса приведено в главах *Создание списка соединений* и *Управление многолистовыми и иерархическими проектами*.

Настройка матрицы правил электрических соединений

Эта матрица позволяет управлять поведением программы проверки правил электрических соединений. Настройка реакции редактора принципиальных схем на определенные условия при тестировании производится установкой цвета соответствующего элемента (рис. 3.40).

Чтение матрицы производится по столбцам и строкам. Например, чтобы понять, какая реакция установлена на условие соединения входного и выходного выводов, нужно найти строку, обозначенную как **Input Pin**, затем отыскать столбец **Output Pin**. Зеленый цвет ячейки матрицы на пересечении выбранных строки и столбца обозначает, что не будет выдано никаких предупреждений при соединении входного и выходного выводов. Таким же образом можно определить с помощью матрицы реакцию на соединение двух выходов разных элементов. В этом случае ячейка имеет красный цвет, что является признаком трактовки программой проверки такой ситуации как ошибочной.

Пользователь имеет возможность определить ошибки или предупреждения для соединений выводов различного типа, портов и входов листа. В общем случае, установленная по умолчанию конфигурация матрицы удовлетворяет основным требова-

ниям пользователя, но при наличии специфических требований матрицу можно легко перенастроить.

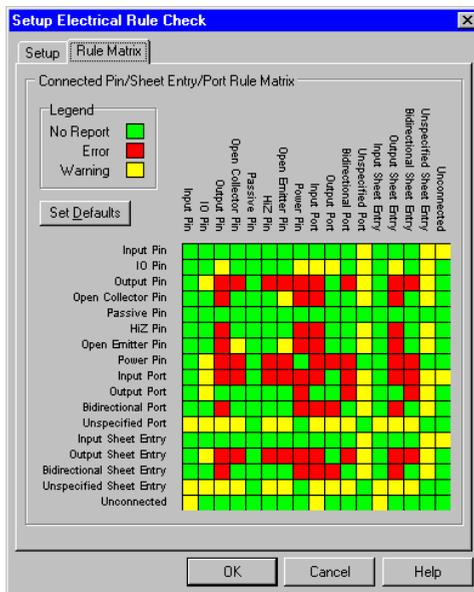


Рис. 3.40. Матрица правил электрических соединений.

Для изменения установленных по умолчанию настроек необходимо последовательно щелкать левой кнопкой мыши на нужной ячейке матрицы. После каждого щелчка цвет ячейки будет изменяться в таком порядке: зеленый (не генерируется никаких сообщений), желтый (генерируется предупреждение) и красный (генерируется ошибка).

Процедура проверки правил электрических соединений является предварительной операцией перед созданием правильного списка соединений проекта. Наличие электрических или логических несоответствий не препятствует созданию редактором принципиальных схем списка соединений, который в этом случае будет неполным или неправильным. Поэтому рекомендуется тщательно проверять и исправлять все ошибки, приведенные в отчете программы проверки, до того, как принять за правильный только что сгенерированный список цепей.

Формат отчета об ошибках

```
Error Report For : C:\CLIENT\SCH3\EXAMPLES\DEM01.ERC    30-Dec-1997
15:18:05
```

```

1 Error Duplicate Designators POWER.SCH C3 At (320,471) And CPU.SCH C3 At
(374,109)
2 Error Multiple Net Identifiers : CPU.SCH RESET At (270,220) And CPU.SCH
RST At (330,220)
3 Warning Unconnected Input Pin On Net N00121
C:\CLIENT\SCH3\EXAMPLES\CPU.SCH(U5-6 310,620)
4 Error Floating Input Pins On Net N00121Pin
C:\CLIENT\SCH3\EXAMPLES\CPU.SCH(U5-6 @310,620)
5 Warning Unconnected Net Label On Net CLOCK CPU.SCH CLOCK
End Report

```

Отчет об ошибках включает информацию о том, на каком листе найдена та или иная бесспорная или вероятная ошибка, координаты ошибки на этом листе, а также соответствующие идентификатор цепи и обозначение компонента.

Исправление ошибок

Редактор принципиальных схем имеет несколько функций, помогающих в процессе исправления ошибок, обнаруженных программой проверки правил электрических соединений.

Просмотр ошибок с помощью браузера

Нижняя половина панели редактора принципиальных схем представляет собой универсальный браузер, используемый для просмотра присутствующих в проекте примитивов, включая маркеры ошибок.

В верхней части браузера необходимо выбрать тип примитива, например, **Error Markers** (маркеры ошибок). Далее следует выбрать из списка интересующую вас ошибку и нажать кнопку **Jump**, после чего маркер этой ошибки появится в центре активного окна.

Следует отметить, что описание ошибки появляется в строке состояния, что во многих случаях помогает избежать постоянного обращения к текстовому отчету. Нажатие на кнопку **Text** в браузере открывает диалоговое окно, которое также может быть использовано для просмотра описания ошибки.

Горячая связь (Cross Probing)

Редактор принципиальных схем поддерживает механизм горячей связи (**cross probing**) с другими открытыми документами. Горячая связь особенно полезна при переходе из среды проектирования в отчет об ошибках. Для того, чтобы подсветить интересующую ошибку или предупреждение в отчете нужно просто щелкнуть мышкой по кнопке **Cross Probe** и затем щелкнуть по соответствующему маркеру ошибки.



Типичные причины возникновения ошибок

Обычно ошибки возникают по следующим причинам:

- ошибки прорисовки — проводники накладываются на выводы, вместо проводников используются графические линии, при выключенной привязке к сетке концы проводников не касаются концов выводов или проводники и шины заканчиваются под портом, вместо того, чтобы касаться конца этого порта.
- ошибки синтаксиса — некорректное написание идентификаторов цепей или неправильные метки шины.
- ошибки компонентов — неправильное размещение выводов компонента или наличие выводов неправильного типа.
- ошибки проектирования — обнаружение программой проверки таких условий проектирования, которые он принимает за ошибочные, например, соединение двух выходов логических элементов.

Трассировка ошибок

Для исправления ошибок нужно начать "обход" меток ошибок, при этом в строке состояния будет отображаться описание ошибок, которые надо рассматривать с учетом приведенных выше причин их возникновения. Если вблизи метки ошибки ничего не заметно, необходимо произвести трассировку этой цепи в проекте. Кнопка **Up/Down Hierarchy** на главной панели инструментов с двумя стрелками, направленными вверх и вниз, помогает при трассировке цепи в многолистовом проекте. В строке состояния выдается краткая инструкция о действиях, которые нужно выполнить при выборе одного из этих инструментов.

Незадействованный вход компонента по существу является признаком незамкнутой цепи. Если сообщается о такой ошибке, значит в каком-то месте цепи, соединяющей этот входной вывод с каким-либо выходным выводом, присутствует разрыв. Если с входным выводом все в порядке, то для обнаружения разрыва необходимо сделать полную обратную трассировку данной цепи.

При обнаружении ошибок во всех цепях какой-нибудь шины искать проблему нужно на уровне этой шины. Причины могут скрываться в неправильном вводе имени порта шины, пропуске метки данной шины или в том, что линия шины заканчивается под портом (что может быть незаметно). Для выявления последней причины нужно

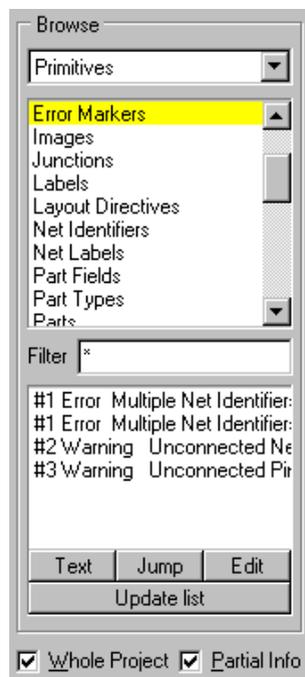


Рис. 3.41. Просмотр ошибок с помощью панели управления редактора схем.



щелкнуть мышью по шине, то есть выделить фокусом, и по положению маркера-манипулятора посмотреть, не заканчивается ли шина под портом.

Подготовка к проектированию печатной платы

После завершения разработки принципиальной схемы пользователь может передать информацию о проекте в редактор топологий, что позволяет начать разработку печатной платы. До того, как сделать это, необходимо изучить данную главу, чтобы окончательно убедиться в том, что вы ничего не упустили.

Обозначение элементов

Все элементы в проекте должны иметь уникальные обозначения. Процесс присваивания и обновления позиционных обозначений называется аннотирование (annotation). Аннотирование может быть выполнено с помощью команды меню **Tools » Annotate** на любой стадии процесса разработки принципиальной схемы. Обновление обозначений можно делать для:

Всех элементов (All Parts)

Обновляются позиционные обозначения всех элементов в проекте. В этом случае все обозначения уникальны.

Элементов заданных шаблоном (? Parts)

Обозначения присваиваются всем элементам, обозначения которых совпадают с заданным шаблоном, например, R?, C?, U? и т. д.

Сброс обозначений (Reset Designators)

Всем обозначениям снова устанавливаются имена R?, C?, U? и т. д., но нумерация начинается с 1.

Если проект имеет сложную иерархическую структуру, необходимо упростить ее и провести обновление позиционных обозначений. Для получения информации о работе с проектами со сложной иерархией обратитесь к пункту *Модель 4 — сложная иерархия* раздела *Управление многолистовыми и иерархическими проектами*.

Группирование секций в составные компоненты

Для того, чтобы определить как идентифицируются и группируются многоэлементные компоненты, например, два отдельных операционных усилителя, расположенных в одном физическом компоненте (корпусе), нужно использовать поля **Group Parts Together If Match By** в диалоговом окне **Annotate**. По умолчанию группировка производится только по полю **Part Type**, для идентификации группы также можно использовать комбинацию из любых 16 полей элементов и 8 библиотечных текстовых полей (рис. 3.42).

Например, есть два операционных усилителя, которые должны быть сгруппированы. Нужно ввести в поле элемента каждого из них строку, идентифицирующую эти элементы. Например, введите строку **Stage1** в поле **Part Field 16** каждого операционного усилителя. После выполнения аннотирования установите флажок **Part Field 16** в **Group Parts Together If Match By**, что обеспечит группирование этих усилителей в одном физическом компоненте.

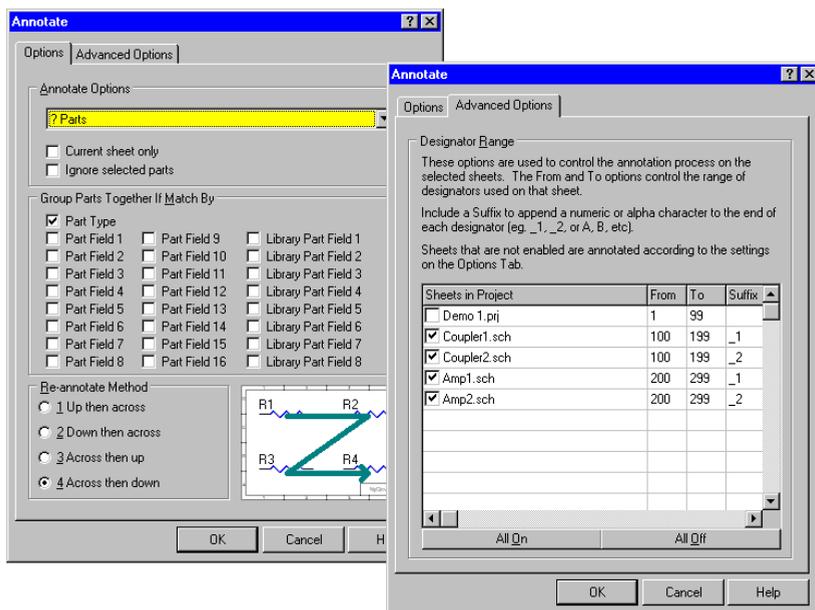


Рис. 3.42. Для управления процессом присвоения позиционных обозначений используются опции, расположенные в диалоговом окне Annotate.

Расстановка позиционных обозначений

Позиционные обозначения присваиваются элементам в зависимости от их положения на листе, которое отсчитывается по зонам, расположенным на краях рамки листа. Для изменения нумерации зон необходимо включить опцию **Use Custom Style** в диалоговом окне **Document Options dialog** и изменить установки **X Ref Region Count** и **Y Ref Region Count**.

Сквозная нумерация элементов в многолистовых проектах

Стиль нумерации позиционных обозначений на принципиальных схемах может быть задан пользователем и применен ко всем листам проекта. Для этого необходимо включить на вкладке **Options** диалогового окна **Annotate** опцию **Current Sheet Only** и перейти на вкладку **Advanced Options**. Здесь представлен список всех листов проекта,

для которых надо включить флаг слева. Ко всем листам, у которых этот флаг не включен, будет применен стандартный стиль, назначенный на вкладке **Options** диалогового окна **Annotate**.

Нумерация позиционных обозначений может быть проведена двумя методами: заданием диапазона численных суффиксов и добавлением специальных суффиксов. В случае задания числовых диапазонов (например, от 1 до 1000) они не должны перекрываться для разных листов, так как в противном случае присвоение уникальных обозначений будет невозможно.

Добавление специальных суффиксов разрешает использование перекрывающихся числовых диапазонов на разных листах. Суффиксы могут быть численными или буквенными. В этом случае нумерация элементов, например, канала 1 будет **R1_1**, **R2_1**, **C1_1** и т. д., а канала 2 будет **R1_2**, **R2_2**, **C1_2** и т. д.

Исключение элементов из процесса автоматической нумерации

Для того, чтобы исключить некоторые элементы из процесса автоматической нумерации позиционных обозначений, их необходимо выделить, после чего включить опцию **Ignore Selected Parts** в диалоговом окне **Annotate**.

Проверка наличия топологических посадочных мест элементов

Для всех элементов проекта должны быть определены топологические посадочные места (то есть, посадочное место с таким именем должно присутствовать в доступной библиотеке топологий (PCB library) при передаче информации из редактора схем в редактор печатных плат).

Простейшим способом проверки наличия топологических посадочных мест элементов является экспортирование информации о посадочных местах в электронные таблицы. Это осуществляется с помощью команды меню **Edit » Export to Spread**, которая вызывает мастер экспортирования принципиальных схем (**Schematic Export Wizard**).

Этот мастер позволяет выборочно экспортировать информацию о принципиальной схеме. Для проверки наличия топологических библиотечных элементов необходимо экспортировать только атрибуты **Designator** (обозначение) и **Footprint** (посадочное место) из списка атрибутов объектов типа **Part**.

Автоматически откроется окно со страницей электронной таблицы, где по пустым полям в столбце **Footprint** можно определить, у каких элементов не заданы топологии посадочных мест. В эти поля можно ввести необходимые названия топологических библиотечных элементов и обновить принципиальную схему с помощью команды меню **File » Update**, появляющейся при работе с окном электронной таблицы. Иначе отредактировать элементы можно непосредственно на принципиальной схеме.

Проверка правил электрических соединений

Обязательным этапом в цикле проектирования является проверка правил электрических соединений, которая позволяет обнаружить в схеме графические и электриче-

ские несоответствия на ранней стадии и предотвратить проблемы, которые могут возникнуть на следующих этапах проектирования. Дополнительные сведения о выполнении проверки электрических правил представлены в разделе *Верификация принципиальных схем*.

Включение в схему топологических директив

Существует возможность включить в принципиальную схему спецификации на печатную плату посредством присоединения к отдельным цепям специальных директив для редактора печатных плат. Для размещения таких директив предназначена команда меню **Place » Directives » PCB Layout**. Когда схема передается в редактор топологий данные об этих директивах преобразуются в топологические правила проектирования. Отметим, что для этого необходимо, чтобы в редакторе печатных плат в диалоговом окне **Update Desing** была включена опция **Generate PCB Rules**. Более подробная информация об этом приведена в разделе *Передача информации о схеме в редактор печатных плат*. После размещения директивы (так, чтобы нижний край символа касался нужной цепи) необходимо отредактировать ее для определения:

- Track Width — ширины проводников этой цепи при трассировке;
- Via Width — диаметра всех переходных отверстий, используемых при трассировке этой цепи;
- Topology — топологии соединений проводников этой цепи;
- Priority — приоритета этой цепи при трассировке;
- Layer — использование слоев при трассировке этой цепи.

Подготовка к проектированию печатной платы

Теперь все готово к тому, чтобы приступить к очередному этапу процесса проектирования — разработке печатной платы. Сначала нужно либо создать пустой PCB документ и затем вручную определить границы (контуры) платы, либо использовать мастер создания печатных плат (**PCB Maker Wizard**), расположенный на вкладке **Wizard** в диалоговом окне **New Document**. Для получения дополнительной информации обратитесь к разделу *Описание печатной платы* главы *Проектирование печатных плат*.

После определения границ печатной платы данные, полученные на этапе разработки принципиальной схемы, могут быть переданы в редактор печатных плат.

Передача информации о схеме в редактор печатных плат

Система Protel 99 SE имеет мощный инструмент синхронизации проекта, который во многом облегчает передачу данных с этапа проектирования принципиальной схемы в редактор печатных плат и обратно.

Синхронизатор автоматически извлекает информацию о компоненте и параметрах связанности из принципиальной схемы, находит соответствующие компоненту топологические посадочные места в библиотеке элементов печатных плат и размещает их на чертеже печатной платы, а затем добавляет линии связи между соединенными выводами компонентов.

Синхронизация проекта представляет собой процесс обновления одного документа (назовем его целевым документом), исходя из последних изменений в другом документе (исходном документе). При запуске синхронизатора нужно задать направление синхронизации (направление передачи данных) выбором одной из двух опций:

- Update PCB — обновление платы, исходя из изменений в схеме
- Update Schematic — обновление схемы, исходя из изменений на плате

Передача информации о схеме

Для передачи данных о принципиальной схеме в редактор печатных плат необходимо выполнить команду **Design » Update PCB** из меню редактора схем, при этом появится диалоговое окно **Update**, в заголовке которого будет присутствовать имя целевого документа.

Не имеет значения, какие листы схемы в проекте в данный момент активны, так как синхронизатор автоматически проанализирует все листы проекта. Если не существует ни одного документа печатной платы, то автоматически будет создан пустой документ.

Проверка предупреждений об ошибках

После выполнения команды **Design » Update PCB** синхронизатор проверяет схему на наличие различных потенциальных ошибок, которые могут привести к проблемам при передаче данных о схеме. Проверка проводится с целью выявления компонентов, не имеющих должного набора обозначений, компонентов с одинаковыми позиционными обозначениями, компонентов с неопределенными (в соответствующей библиотеке) топологическими посадочными местами. Также проверяется подключение к редактору печатных плат необходимых библиотек топологических элементов.

Если такие потенциальные ошибки обнаружены, то в диалоговом окне **Update** добавляется дополнительная вкладка **Warning**, содержащая предупреждения о найденных ошибках. Перед выполнением синхронизации необходимо просмотреть содержимое этой вкладки. Для получения полной информации о предупреждениях нужно нажать

кнопку Report, расположенную в нижней части вкладки **Warning**. Способы исправления некоторых из ошибок, например, обновление позиционных обозначений, поиск и присвоение пропущенных топологических элементов описаны в предыдущем разделе *Подготовка к проектированию печатной платы*.

Диалоговое окно синхронизатора проекта

Диалоговое окно синхронизатора проекта **Update Design** (рис. 3.43) включает следующие поля:

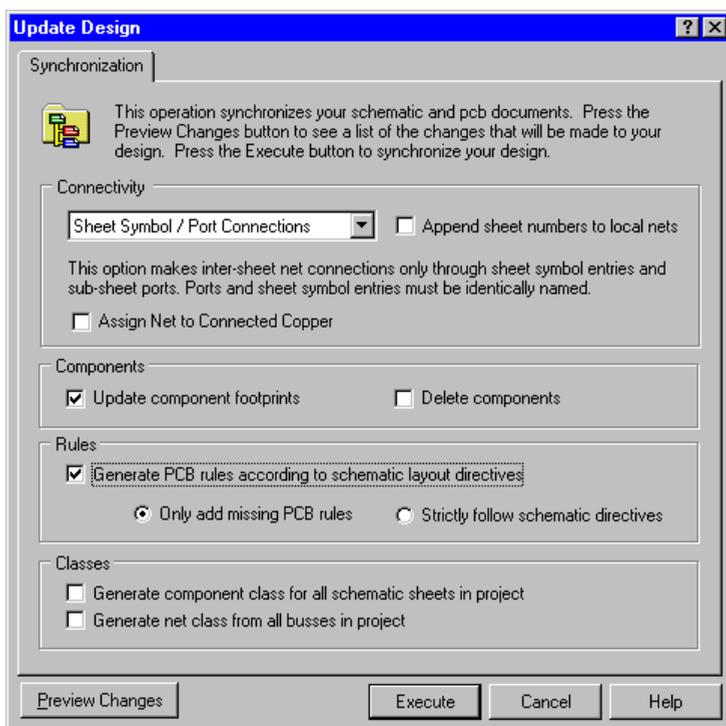


Рис. 3.43. Настройка синхронизатора проекта.

Connectivity (связанность)

Здесь присутствует выпадающий список **Net Identifier Scope**, задающий область видимости идентификаторов цепей. Выбранный элемент этого списка определяет, каким образом создается связанность между листами в многолистовом проекте. Если для связи с подчиненными листами в проекте символы листа (**sheet symbols**) используют входы листа (**sheet entries**), то из этого списка нужно выбирать опцию **Sheet Symbols/Port Connections**. Если в проекте для прямого соединения между подчиненными листами используются цепи или порты, необходимо выбрать какой-либо другой ва-

риант из списка **Net Identifier Scope**. Наиболее полная информация по вопросу связанности листов приведена в разделе *Управление многолистовыми и иерархическими проектами*.

Опция **Assign Net to Connected Copper** включает режим, когда всем элементам трассировки (проводникам, переходным отверстиям) присваивается имя цепи, которое затем проверяется с соответствующими именами контактных площадок. Это полезно делать при внесении изменений в схему, так как после этого для поиска обновленных цепей в редакторе печатных плат вы сможете использовать проверку правил проектирования. Операция присвоения выполняется с помощью команды **Design » Netlist Manager** меню редактора печатных плат.

Components (компоненты)

В этом поле два флага. Установка флага **Delete Components** (удалить компоненты) приведет к автоматическому удалению всех находящихся в целевом документе компонентов, которые не имеют соответствующего им компонента в исходном документе. Это очень удобно в случае, когда в принципиальную схему было внесено большое количество изменений, и теперь хотелось бы автоматически удалить все лишние компоненты на печатной плате.

Второй флаг — **Update Footprints**. Установка этого флага приведет к передаче изменений данных о топологических посадочных местах из исходного документа в целевой документ.

Rules (правила проектирования)

Как описывалось ранее, принципиальная схема может содержать некоторые данные о печатной плате в специальных топологических директивах (**PCB Layout directives**). Синхронизатор может передавать такие данные в редактор печатных плат для генерации соответствующих правил проектирования, для чего необходимо установить флаг **Generate PCB Rules**. У этой функции существует два режима:

Only add missing PCB rules (добавление только отсутствующих правил) — для цепей, которые не имеют никаких правил, создаются новые правила; существующие правила обновляются исходя из данных в топологических директивах.

Strictly follow Schematic directives (строгое следование директивам) — для цепей, которые не имеют никаких правил, создаются новые правила; существующие правила обновляются исходя из данных в топологических директивах; любые другие правила проектирования для цепей, попадающих в область видимости, удаляются.

Для получения более полной информации о передаче данных из топологических директив для генерации правил проектирования обратитесь к разделу *Передача топологической информации* данной главы.

Classes

Опция **Generate Component Classes** позволяет создавать классы объектов на чертеже печатной платы исходя из наборов компонентов на листах схем. Имя класса получается из имени листа принципиальной схемы посредством удаления из него пробелов. Если различные секции составного компонента располагаются на разных листах, то этот компонент включается в класс того листа, где расположена его первая секция.

Для каждого класса компонентов в последствии будут созданы области (room) размещения, которые можно будет размещать на плате. Более подробная информация по этому вопросу приведена в подразделе *Работа с областями размещения* главы *Проектирование печатных плат*.

Опция **Generate Net Classes from Buses** позволяет в редакторе печатных плат создавать классы цепей для имеющихся на схеме шин.

Предварительный просмотр вносимых изменений

В нижней части диалогового окна **Update** расположена кнопка **Preview Changes**. Нажатие на эту кнопку открывает дополнительную вкладку **Changes**, где в виде списка перечислены все изменения, которые собирается сделать синхронизатор в целевом документе (рис. 3.44).

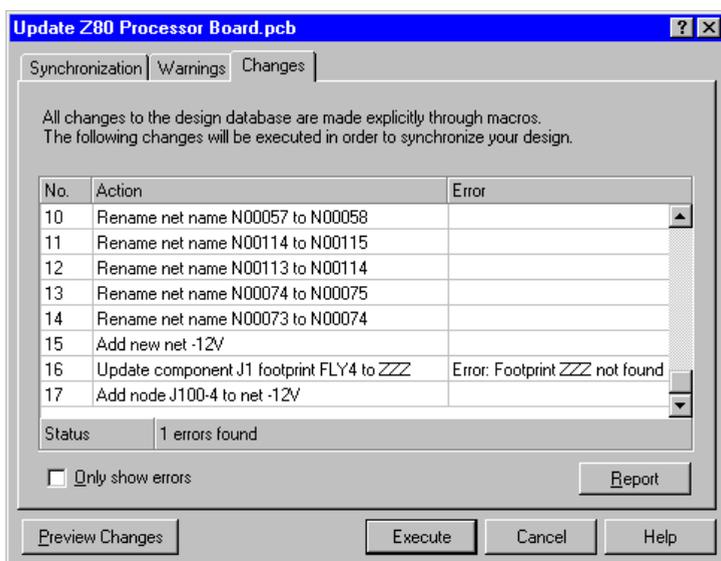


Рис. 3.44. Предварительный просмотр сделанных изменений дает возможность выявить и исправить возможные ошибки

В нижней части вкладки расположен флажок **Only show errors**, с помощью которого можно сократить список, оставив в нем только ошибки. По необходимости из этого списка можно удалить любую операцию, щелкнув по ней правой кнопкой мыши и в появившемся контекстном меню выбрав команду **Delete** (удалить). Кнопка **Report** генерирует полный отчет обо всех приведенных в списке операциях.

Как синхронизатор определяет, какой чертеж печатной платы использовать?

При определении того, какой лист (или листы) печатной платы связан с каким листом (листами) принципиальной схемы синхронизатор использует следующую стратегию. В первую очередь он ищет лист печатной платы в той же папке базы данных проекта (Design Database), где располагается данная принципиальная схема. Если поиск завершился удачно, то используется найденный лист. Если в этой папке имеется несколько разных чертежей печатных плат, то выдается запрос на выбор правильного. Если поиск в папке завершился неудачно, то просматривается вся база данных проекта, составляется список всех обнаруженных чертежей печатных плат и выдается запрос на выбор одного из них. В случае отсутствия требуемых чертежей во всей базе данных создается новый проект печатной платы в той же папке, где располагается лист принципиальной схемы.

Как размещаются компоненты на чертеже печатной платы?

Синхронизатор размещает топологические посадочные места компонентов в ряд по горизонтали. Если до передачи данных из схемы был определен контур размещения на печатной плате в специально отведенном для этих целей слое, то такие компоненты размещаются в правой части этого контура. В иных случаях компоненты размещаются в абсолютном центре чертежа с координатами 50000, 50000.

Передача топологической информации

Еще одна функция, которую выполняет синхронизатор, это трансляция топологических директив, расположенных на принципиальной схеме, в правила проектирования печатной платы. Настройки каждой директивы транслируются в соответствующие правила с учетом области видимости цепей. Заметим, что топологические директивы работают только с дюймовой системой единиц, что доставляет много неудобств пользователям, работающим в редакторе печатных плат с метрической системой.

Таблица 3.2

Атрибуты печатной платы	Правила проектирования
Track Width (ширина проводника)	Width Constraint (ограничение на ширину)
Via Width (ширина переходного отверстия)	Routing Via Style (стиль переходных отверстий)
Topology (топология)	Routing Topology (топология при трассировке)
Priority (приоритет)	Routing Priority (приоритет при трассировке)
Layer (слой)	Routing Layers (слои для трассировки)

Передача изменений в принципиальной схеме в редактор печатных плат

Обычный ход процесса проектирования — это внесение изменений в разрабатываемую схему с последующей передачей этих изменений в уже существующий проект печатной платы.

Это делается точно так же, как и самая первая передача данных из принципиальной схемы в чертеж печатной платы, то есть с помощью команды меню **Design » Update РСВ**. При этом синхронизатор анализирует схему и плату с целью поиска отличий. Затем он создает набор макросов — процедур, устраняющих каждое отдельное отличие, и запускает их на выполнение.

Как синхронизатор сопоставляет компоненты на принципиальной схеме и на печатной плате?

При синхронизации принципиальной схемы с печатной платой согласующимся друг с другом компонентам присваиваются идентификаторы соответствия. Такой подход избавляет от необходимости повторного аннотирования отдельно схемы и платы, которые могут быть согласованы в любое время с помощью запуска команды меню **Update**.

Если в схеме или на плате будет найден компонент без идентификатора соответствия, то синхронизатор попытается подобрать для него подходящий компоненты и после этого выведет диалоговое окно **Confirm Components Associations** для подтверждения этого соответствия (рис. 3.45).

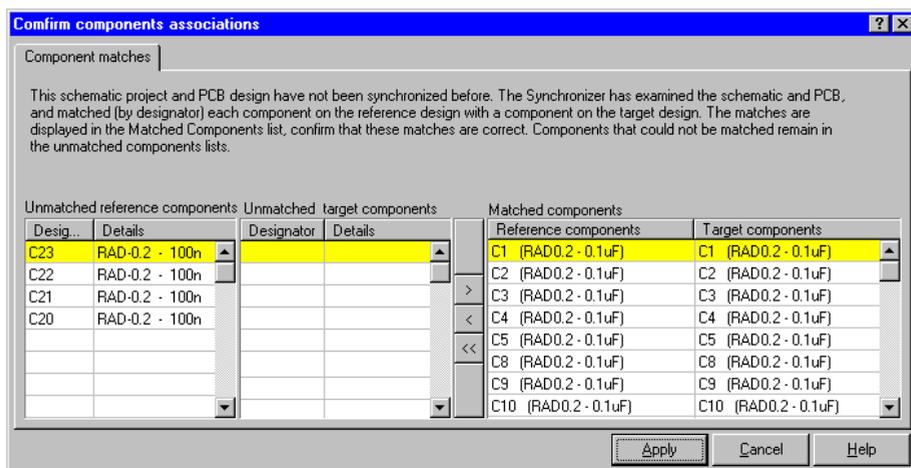


Рис. 3.45. Согласующиеся и не согласующиеся компоненты отображаются в различных списках диалогового окна Confirm Components Associations.

Синхронизатор выполняет этот предварительный подбор с использованием позиционных обозначений компонентов, поэтому необходимо всегда следить за его правильностью. При нажатии на кнопку **Apply** подобранные компоненты получают идентификатор соответствия.

Если обновляется печатная плата, исходя из изменений в принципиальной схеме, то несогласующиеся компоненты, расположенные на схеме, добавляются на плату, а несогласующиеся компоненты, расположенные на плате, либо удаляются, либо выносятся в отчет, в зависимости от состояния опции **Components** в диалоговом окне **Update**.

Если обновляется принципиальная схема, исходя из изменений в печатной плате, то несогласующиеся компоненты, расположенные на плате, добавляются в список в **Preview Changes Report** (предварительный просмотр отчета об изменениях), а несогласующиеся компоненты, расположенные на схеме, либо удаляются, либо выносятся в отчет, в зависимости от состояния опции **Components** в диалоговом окне **Update**.

Как синхронизатор передает проектную информацию

Для передачи проектной информации из схемы на плату синхронизатор извлекает информацию о компоненте и связанности, после чего создает набор макросов. Каждое действие, которое должно быть выполнено, например, добавление нового компонента, добавление новой цепи или добавление узла цепи, определяется соответствующим макросом.

Если макрос по какой-либо причине не может быть выполнен (например, из-за отсутствия в библиотеке нужной топологии посадочного места), то он будет добавлен в качестве ошибки в список, расположенный на вкладке **Change** в диалоговом окне **Update**.

Исправление ошибок выполнения макросов

Перед выполнением макросов всегда рекомендуется исправлять ошибки и предупреждения. Далее приведено описание ошибок и предупреждений, а также поясняется, при выполнении каких макросов могут появиться эти ошибки, и что является причиной их появления.

Net not found (цепь не найдена)

Макрос производит одно из следующих действий: добавляет или удаляет узел, удаляет цепь, изменяет имя цепи, тогда как такой цепи нет в списке соединений печатной платы.

Component not found (компонент не найден)

Макрос производит одно из следующих действий: добавляет или удаляет узел, когда обозначение компонента в макросе определено не правильно, или этого компонента нет в списке соединений печатной платы; удаляет компонент; изменяет посадочное место, обозначение или комментарий для компонента, которого нет в списке соединений печатной платы.

Node not found (узел не найден)

Макрос производит одно из следующих действий: добавляет или удаляет узел к несуществующему выводу компонента или несуществующей цепи.

Net already exists (цепь уже существует)

Макрос пытается добавить имя цепи, хотя цепь с таким именем уже существует в списке соединений платы.

Component already exists (компонент уже существует)

Макрос пытается добавить компонент, обозначение которого совпадает с уже существующим в списке цепей платы компонентом.

New footprint not matching old footprint (новое топологическое посадочное место не совпадает со старым)

Макрос пытается изменить посадочное место компонента, тогда как задействованные выводы старого топологического элемента не совпадают с задействованными выводами нового. Такая ситуация может возникнуть, если новый компонент имеет меньше выводов, чем старый, или если нумерация выводов компонента, взятого из принципиальной схемы, отлична от нумерации выводов того же компонента на печатной плате.

Footprint not found in Library (в библиотеке не найден указанный топологический элемент)

Макрос пытается добавить новый компонент или изменить топологическое посадочное место компонента, когда нужная топология не найдена ни в одной подключенной библиотеке и нет ссылок на альтернативные библиотеки в файле перекрестных ссылок (**Cross Reference**) **ADVPCB.XRF**.

Alternative footprint used instead (используется топологический элемент из альтернативной библиотеки)

Это предупреждение генерируется, когда макрос пытается добавить новый компонент или изменить топологическое посадочное место компонента, когда указанная топология не найдена ни в одной подключенной библиотеке. Однако, в файле перекрестных ссылок (**ADVPCB.XRF**) была найдена ссылка на альтернативную библиотеку и компонент будет загружен из одной из библиотек, находящихся в списке доступных. Рекомендуется всегда проверять и подтверждать правильность выбранного альтернативного топологического посадочного места, перед тем, как выполнить макрос с таким предупреждением.

Заметим, что когда макрос пытается загрузить или изменить посадочное место компонента, которое не обнаружено ни в одной библиотеке, он для дальнейшего поиска в файле перекрестных ссылок использует графу примечание (**Comment**). Файл перекрестных ссылок содержит сортированный по типу список компонентов и соответствующие им топологические посадочные места. Например, если компонент U1 это микросхема 74LS00, а пользователь забыл включить его топологию при создании макроса, который добавляет этот компонент, тогда макрос будет искать 74LS00 в файле перекрестных ссылок и найдет, что эта микросхема имеет посадочное место DIP14, которое и будет загружено из одной из подключенных библиотек.

Резюме

Большинство проблем, связанных с передачей данных из проекта принципиальной схемы в редактор печатных плат, в основном делится на две категории.

1. Потерянные топологические элементы. Потери топологий компонентов происходят в следующих случаях: отсутствие информации о топологических посадочных местах компонентов в списке соединений; отсутствие требуемых библиотек элементов печатных плат в списке подключенных библиотек; отсутствие данных о топологии во всех доступных библиотеках.
2. Новые топологические посадочные места не соответствуют старым. Обычно причина кроется в том, что нумерация выводов компонента, взятого из принципиальной схемы, отлична от нумерации выводов того же компонента на печатной плате.

Библиотеки элементов принципиальных схем содержат специфические компоненты и устройства. Библиотеки топологических элементов печатных плат содержат общие для некоторого набора компонентов топологии, которые могут принадлежать различным специфическим компонентам, каждый из которых может иметь разную нумерацию выводов.

Например, разные транзисторы могут иметь разные выводы базы, эмиттера и коллектора, и эта информация должна быть правильно передана в редактор печатных плат. То же самое относится к диодам, которые имеют выводы "А" и "К".

Поэтому возникает необходимость либо изменить количество выводов топологического посадочного места, чтобы оно соответствовало количеству выводов элемента на схеме, либо наоборот.

Печать принципиальных схем

Обзор

Ввод принципиальной схемы является только частью процесса проектирования. В большинстве случаев необходимо получить изображение схемы, которое можно сохранить и в дальнейшем просмотреть. Для этих целей редактор принципиальных схем включает поддержку разнообразных опций для получения твердой (печатной) копии схемы. Фактически любое устройство, которое поддерживается Windows, может быть использовано для распечатки схем проекта.

Вывод на принтер или плоттер

Вывод схем на принтер или плоттер в системе Protel 99 SE ничем не отличается от распечатки документов в любом другом приложении Windows. Операционная система Windows управляет процессом печати и обеспечивает его необходимым набором драйверов: растровых и PostScript для печати на принтерах и векторных для вывода на плоттеры. В этот набор входят драйверы поддержки как старых девятигольчатых

матричных принтеров, так и современных компьютерных фотонаборных устройств высокого разрешения, а также драйверы для разного вида плоттеров.

- ◆ Вращение шрифтов не поддерживается, поэтому замена шрифтов может быть использована только в том случае, если текст на схеме находится в стандартном горизонтальном (или книжном) положении и размер текста не превышает максимально возможный для принтера размер. PostScript принтеры поддерживают вращение шрифтов под любым углом.

Для настройки параметров печати из активного окна редактора схем или редактора библиотек нужно использовать команду меню **File » Setup Printer** (горячие клавиши **F, R**).

Следующий раздел описывает опции, представленные в диалоговом окне настройки печати (рис. 3.46).

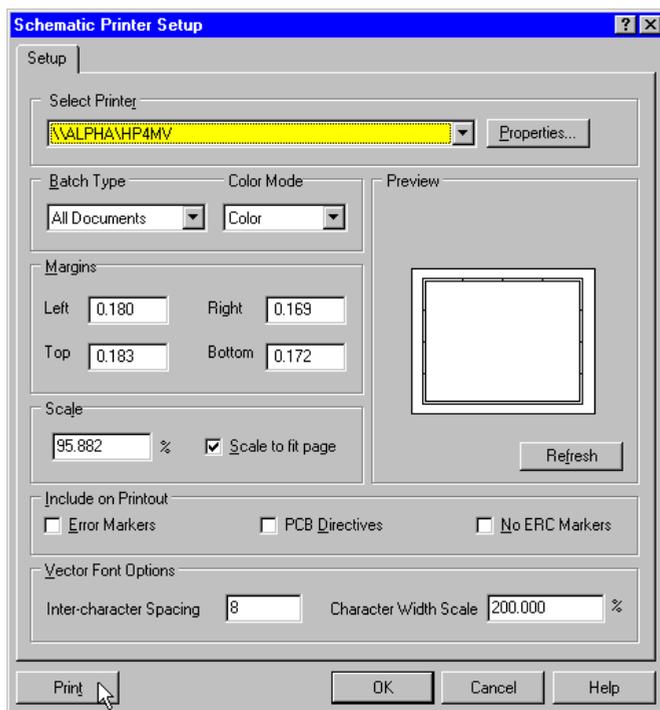


Рис. 3.46. Окно настройки печати принципиальных схем.

Selecting a Printer (выбор принтера)

Это выпадающее меню содержит все устройства вывода, которые были установлены с помощью Панели управления Windows (см. Руководство пользователя Microsoft Windows). Эти устройства поддерживаются стандартными драйверами, поставляющимися с операционной системой, или драйверами производителей этих устройств. Нужно отметить, что новые и обновленные драйверы для новых и уже существующих устройств выпускаются регулярно. За последней информацией о принтерных драйверах обращайтесь в службы поддержки Microsoft Windows и производителей устройств печати.

Batch Type (пакетная печать)

С помощью этой опции можно определить один или все открытые листы будут выведены на печать в редакторе схем (включая все открытые проекты). Если эта опция используется в редакторе библиотек, то выбранный вариант определяет печать одного редактируемого в данный момент компонента или всех компонентов, содержащихся в текущей библиотеке. Последняя опция позволяет распечатать полную библиотеку компонентов всего лишь за одну операцию. При выборе этой опции распечатываются все изображения компонентов, включая их начертания в форматах DeMorgan и IEEE, когда они существуют, а также информация из текстовых полей описания компонентов. Кроме того, эта опция учитывает все другие опции печати, например, масштабирование.

Color mode (режим цвета)

В этом выпадающем списке задается цветной (**Color**) или монохромный (**Monochrome**) режим вывода на печать. При выборе цветного режима распределение экранных цветов на схеме используется для назначения цветов при печати или выводе на плоттер, основанных на опциях, которые доступны в драйвере печати. Количество оттенков серого и распределение цветов по шкале зависит от драйвера и самого устройства печати. В монохромном режиме печать производится только черным цветом. Поддержка эффекта размывания и печать оттенков серого в этом режиме не предусмотрена. Монохромный режим рекомендуется использовать для печати на матричных принтерах с низким разрешением, а также на устаревших плоттерах с одним пером.

Margins (отступы)

Пределы управления полями печатного листа ограничено только размерами отступов, заложенными в принтер или плоттер, которые не позволяют выводить изображение на краях листа (например, принтеры PostScript). При использовании отступов с опциями **Scale** (жестко заданный масштаб) и **Scale to Fit Page** (автоматический масштаб под размер страницы) размер печатной области ограничивается только установленными значениями полей листа и сохранением геометрических пропорций изображения. Для того, чтобы установить максимальную область печати нужно во всех полях отступов ввести нулевые значения и нажать кнопку **Refresh**. При этом параметры отступов автоматически примут минимальные значения для выбранного принтера.

Масштабирование печати

Печать на принтере и вывод на плоттер может производиться с масштабированием изображения. При этом коэффициент может быть определен вручную или автоматически с учетом предустановленных полей страницы.

Scale (жестко заданный масштаб)

В этом поле вводится коэффициент масштабирования от 0.01% до 400%.

Scale to Fit Page (автоматический масштаб)

При включении этой опции и выводе на печать изображение автоматически масштабируется таким образом, чтобы максимально заполнить страницу с учетом установленных размеров полей и ограничений выбранного для печати принтера и сохранением геометрических пропорций.

При применении автоматического масштабирования под размер страницы можно устанавливать нулевые значения отступов, тогда драйвер принтера автоматически учтет наличие и размер непечатаемых областей.

Tiling (разбиение большой схемы для печати на нескольких листах малого размера)

Если размер подготовленного для печати документа, например, листа принципиальной схемы или библиотеки, превышает доступную для выбранного устройства область печати, редактор схем автоматически распределит изображение на два или более листов. На таких листах правильные отступы выдерживаются на внешних границах каждого листа. Для предварительного просмотра результата такого размещения нужно нажать кнопку **Refresh**.

Очень часто существует возможность уменьшить количество печатаемых листов. Это можно сделать с помощью изменения ориентации листов и настройкой отступов. Для предварительного просмотра можно использовать окно **Preview** в диалоговом окне **Printer Setup**.

Properties (настройка принтера)

Нажатие на кнопку **Properties** вызовет диалоговое окно **Print Setup**, где задаются все доступные для выбранного принтера параметры и опции. В зависимости от устройства эти опции могут включать: размер и ориентацию листа, какой лоток используется принтером для подачи бумаги и т.п. Посредством этого окна можно получить доступ к собственному диалоговому окну настройки принтера.

Refresh (обновить)

После изменения каких-либо параметров настройки можно нажать кнопку **Refresh** для обновления окна предварительного просмотра распечатки.

Печать

После установки всех необходимых настроек нужно нажать либо кнопку **Print** для запуска процесса печати, либо кнопку **OK** для сохранения всех сделанных изменений,

либо кнопку **Cancel** для выхода из окна **Printer Setup** без сохранения новых параметров. Когда генерируется образ печатного листа (либо непосредственно на устройстве вывода, либо в файле) текущая распечатываемая страница или слой отображается в диалоговом окне. Если печать производится в файл, то пользователь по запросу программы должен ввести имя выходного файла.

Печать на PostScript принтере

Некоторые PostScript принтеры имеют определенное значение времени ожидания и, если за это время не получают маркер конца страницы, то отбрасывают текущие данные. Это может стать причиной пропуска страниц. Устранить эту проблему можно следующим образом: откройте Панель управления (**Control Panel**), щелкните по иконке Принтеры, выберите принтер и нажмите кнопку конфигурирования, затем введите в поле повтора передачи (**Transmission Retry**) некоторое достаточно большое число, например, 500 секунд.

В некоторых случаях изображение на отпечатанной странице может быть не полным, например, напечатаны все компоненты, но нет проводников. Такая ситуация может сложиться из-за нехватки памяти в принтере. Лазерные принтеры размещают все изображение в свою память до того, как начать процесс печати, поэтому если изображение не помещается в памяти, возможна потеря отдельных элементов изображения, а распечатано будет только содержимое памяти принтера.

Создание отчетов

Спецификация материалов (Bill of Materials)

При выборе команды **Report » Bill of Materials** вызывается Мастер по созданию спецификации материалов принципиальной схемы (**Schematic Bill of Materials (BOM) Wizard**). Ниже описаны инструкции по каждой вкладке этого мастера. Отчет по спецификации может быть представлен одним из трех форматов:

Protel Format (формат фирмы Protel)

Файл в этом формате представляет собой текстовый файл в формате ASCII расширением **BOM**, где столбцы разделены символом табуляции. Файл содержит список полей типов элементов, количество элементов каждого типа и поля обозначений, присвоенных каждому типу.

CSV Format (формат с разделением запятой)

Отчет в формате **CSV (Comma Separated Value;** значения, разделяемые запятой) включает полное описание каждого компонента, как на уровне схемы, так и на библиотечном уровне. Файл отчета имеет расширение **CSV**.

Client Spreadsheet (пользовательские электронные таблицы)

Включает полное описание каждого компонента, как на уровне схемы, так и на библиотечном уровне. Файл в этом формате автоматически загружается в редактор пользовательских электронных таблиц, откуда может быть сохранен в формате электронных таблиц системы Excel.

Перекрестные ссылки (Cross Reference)

Этот текстовый отчет в формате ASCII с расширением **XRF** содержит список обозначений элементов, типов и расположение листов (имена файлов) для каждого элемента.

Иерархия проекта (Design Hierarchy)

Этот текстовый отчет в формате ASCII с расширением **REP** содержит список всех файлов активного проекта.



Рис. 3.47. Настройка горячей связи с внешней базой данных.

Сравнение списков соединений (Netlist Compare)

Этот текстовый отчет в формате ASCII с расширением **REP** содержит перечень отличий двух списков соединений. Этот отчет используется для отслеживания изменений в документе, сделанных при переходе от одной версии проекта к другой. Эта функция работает с форматами списка соединений форматов **Protel**, **Protel2** и **Tango**. В отчет включаются сходные цепи, частично сходные цепи, добавочные цепи в первом и во втором списке соединений и общее число цепей в каждом списке соединений.

Связь с внешними базами данных

Редактор принципиальных схем поддерживает два мощных и гибких метода связи с внешними базами данных.

- Горячая связь (**hot linking**) полей элемента с внешними базами данных. Эта функция позволяет вносить информацию из внешней базы данных непосредственно в текстовые поля элементов, расположенных на листе схемы.
- Функции импорта и экспорта баз данных. Они позволяют экспортировать и импортировать значения различных атрибутов и объектов, размещаемых в редакторе принципиальных схем.

Горячая связь с базой данных

Горячая связь с базой данных позволяет объединить в реальном времени информацию из внешней базы данных с полями элементов, используемых в принципиальной схеме.

Созданные связи с внешней базой данных не принадлежат какому-либо отдельному компоненту или листу, они относятся ко всем компонентам на всех открытых в редакторе листах схем. Эти связи сохраняются в файле настроек редактора схем, имеющем расширение **INI**.

Для установления связей необходимо с помощью команды меню **Option » Database Links** вызвать диалоговое окно **Linking Setup** (настройка связи). Любое из шестнадцати текстовых полей элементов может быть присоединено к базе данных. Все они могут быть присоединены к одной базе или к разным базам данных. Для активизации связи элемента нужно установить флаг рядом с именем этого элемента. После активизации связей редактор схем будет делать попытки прочитать информацию из внешней базы данных и обновить соответствующее поле элемента. Время цикла обновления задается в поле **Update Every** в минутах (рис. 3.47).

Задание связей

Для задания связей между текстовым полем элемента и внешней базой данных нужно в диалоговом окне **Linking Setup** выбрать нужное поле **Part Field** и нажать кнопку **Configure**. Появится диалоговое окно **Database Linking for Part Field**.

Определение положения базы данных

Первый шаг — выбор базы данных, с которой надо установить связь. Для этого нужно ввести с клавиатуры имя файла базы данных и полный путь к ней или воспользоваться кнопкой **Browse**. Поддерживаемые форматы баз данных включают dBase III и dBase IV.

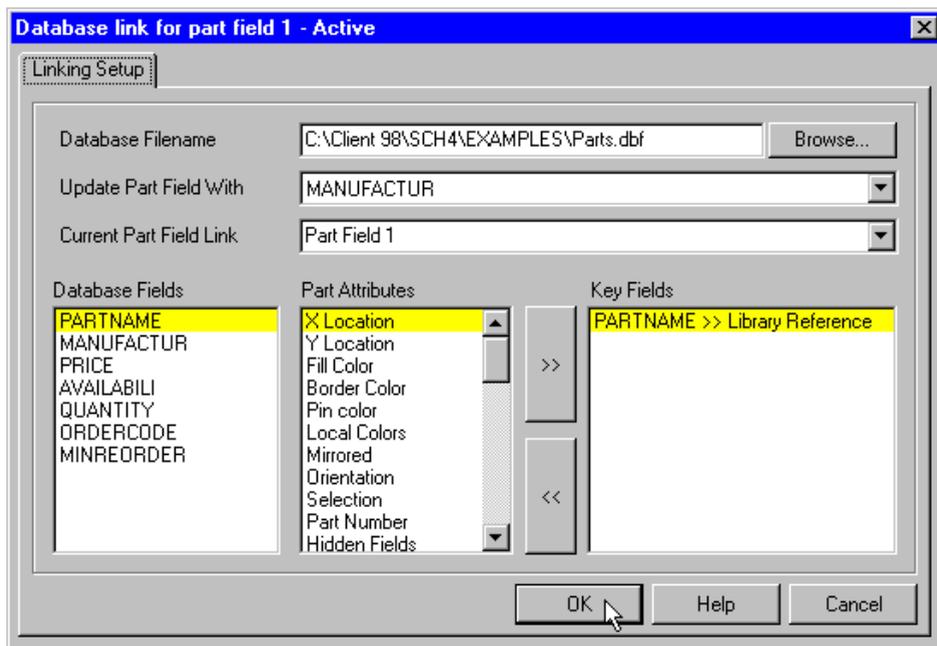


Рис. 3.48. Задание связи между текстовым полем элемента и базой данных.

Выбор поля базы данных

После выбора требуемой базы данных редактор схем может прочитать в ней все доступные поля. Следующим шагом является выбор поля в базе, которое нужно отобразить на поле конфигурируемого в данный момент элемента. Поле базы данных выбирается из выпадающего списка **Update Part Field With**. В примере, показанном на рисунке поле **MANUFACTUR** отображается в **Part Field 1**.

◆ Обратите внимание, что в заголовке окна **Database Link** присутствует признак **Activate** или **Inactivate**, показывающий установлен ли флаг для текущего поля элемента в окне **Linking Setup**.

При совпадении данных в ключевых полях (PATRNAME=Lib Ref) вся информация из базы данных переносится в назначенные текстовые поля. В данном примере для простоты имена полей выбраны одинаковыми.

PATRNAME	MANUFACTUR	PRICE	AVAILABILI	QUANTITY	ORDERCODE	MINREORDER
6264	Texas Instruments	\$1.00	3 Days	30	T-47832	15
2764	Intel	\$2.00	1 Day	30	I-875543	10
Z80ASIO	Intel	\$4.00	2 Weeks	5	I-890765	2

Обратите внимание, что все имена текстовых полей компонента были изменены в редакторе библиотек таким образом, чтобы было легче разобраться в их содержимом.

Рис. 3.49. Принцип действия горячей связи с внешней базой данных.

Назначение ключевых полей

Целью назначения ключевых полей является определение того, как идентифицировать нужную запись во внешней базе данных. Разные поля элементов может быть связаны с разными базами данных, а также могут использовать разные ключевые поля.

Например, рассмотрим базу, которая содержит список компонентов, где каждая запись состоит не только из названия компонента, но также и из дополнительных полей, таких как производитель, цена, наличие и др. В примере, приведенном на рисунке 3.49, содержимое поля базы данных **PARTNAME** используется как идентификатор записи, который настроен на ссылку компонента схемы в библиотеке так, что связь устанавливается, когда их содержимое совпадает. Как только такая связь между записью в базе данных и полем элемента обозначилась, содержимое любого поля этой записи из базы может быть извлечено и загружено в поле элемента.

Для того, чтобы назначить ключевое поле нужно выбрать поле в выпадающем списке полей базы данных (в **Database Fields** выбрать **PARTNAME**), выбрать атрибут элемента, в котором оно будет отображаться (в **Part Attributes** выбрать **Library Reference**) и затем нажать кнопку >> для пополнения списка ключевых полей (смотри список **Key Fields** в диалоговом окне **Database Linking for Part Field**).

Для успешного установления связи с внешней базой данных содержимое ключевых полей должно быть полностью идентичным. Например, при назначении поля с именем **PARTNAME** на атрибут схемного компонента **LibReference** для установления связи между ними содержимое **PARTNAME** должно полностью совпадать с содержимым атрибута **LibReference**. Только в этом случае содержимое любого поля из записи базы может быть передано в соответствующее поле элемента. Это особенно важно, если в ключевых полях используются символы кириллицы, так как система Protel 99 SE импортирует кириллицу в кодировке Win1251, а база может иметь кодировку DOS.

Обновление данных

Связь между базами данных и полями элементов является динамической, то есть информация обновляется по мере изменения ее в базе данных. Этот процесс выполняется автоматически через заданный промежуток времени. Выключить автоматическое обновление для какого-либо поля, то есть сделать связь этого поля с базой данных неактивной (**Inactive**), можно сбросив флаг активности для этого поля в диалоговом окне **Linking Setup**.

Импорт и экспорт баз данных

Каждый объект (или примитив), который может быть размещен в редакторе принципиальных схем, имеет некоторый набор атрибутов. Например, проводник имеет три атрибута: цвет, ширину и флаг выделения. Компоненты имеют 33 атрибута. Функции импорта и экспорта позволяют выбирать те атрибуты, значения которых должны быть переданы в или получены из базы данных. Также можно определить область действия этих функций, например, текущий лист, текущий проект или все открытые листы.

Как только значения выбранных атрибутов экспортируются в базу данных, они могут быть использованы в любой СУБД или в любом программном продукте по обработке электронных таблиц, которые могут осуществлять операцию чтения базы данных в выбранном формате. Поддерживаемые форматы включают dBase III и dBase IV. СУБД может также использовать свои функции по редактированию этой базы данных.

Для того, чтобы иметь возможность успешно экспортировать данные в базу, отредактировать их и затем импортировать обратно, нужно иметь некоторый механизм уникальной идентификации каждого экземпляра какого-либо примитива, расположенного в области действия функции импортирования.

Специальные ключевые атрибуты

Для идентификации любого примитива на листе и того, к какому листу он принадлежит, примитивы имеют специальные атрибуты положения (**location**). Существует два специальных атрибута положения. Каждый примитив имеет атрибут имени файла документа (**Document File Name**) для идентификации листа, на котором он находится. Примитивы, у которых нет информации о собственных координатах, например, проводники или линии, дополнительно имеют атрибут массив вершин (**Vertex Array**). Этот атрибут содержит массив вершин данного проводника или линии.

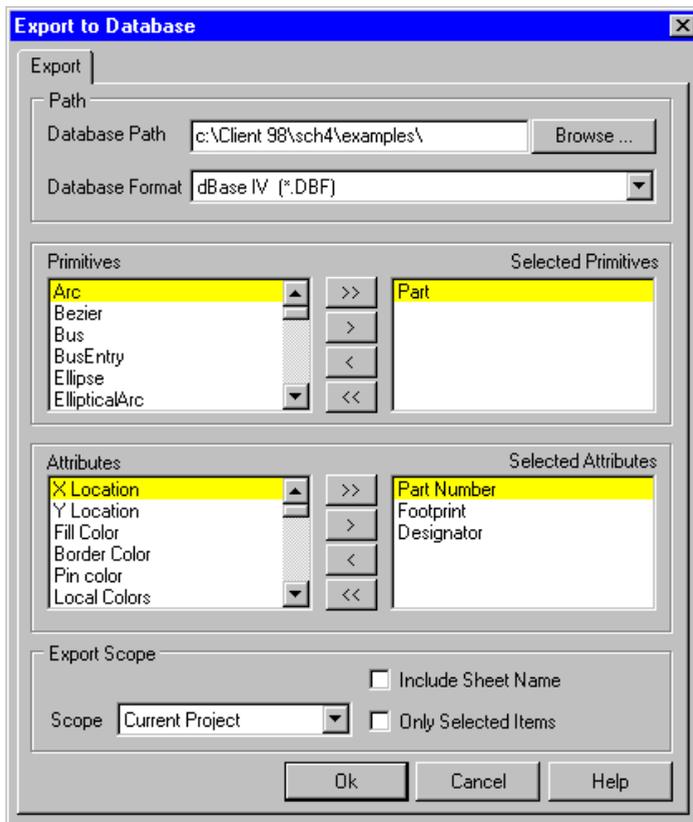


Рис. 3.50. Настройка экспорта в базу данных.

При импортировании из базы данных эти атрибуты положения используются в качестве ключевых атрибутов (**Key Attributes**), которые программа использует для идентификации каждого экземпляра графического примитива при загрузке данных из базы.

Экспорт в базу данных

Для вызова диалогового окна **Export to Database** (рис. 3.50) нужно выполнить команду меню **File » Export to Database**.

Имя по умолчанию для этой базы данных будет предложено на основании выбранного примитива.

Выбор примитивов

Окно со списком графических примитивов (окно **Primitives**) обеспечивает доступ к любому примитиву или объекту, имеющемуся в редакторе принципиальных схем. Нужные примитивы выбираются из этого списка и добавляются в список выбранных примитивов (**Selected Primitives**).

После выбора примитива в списке **Attributes** нужно указать, какие из его атрибутов будут экспортироваться. Эти действия нужно выполнить для каждого конкретного объекта.

Каждый тип атрибута становится отдельным полем ячеек в базе данных, имя которого назначается автоматически при выполнении операции экспорта.

Выбор атрибутов

Для каждого выбранного примитива нужно указать, какие из его атрибутов должны быть экспортированы. Если позднее будет осуществляться импорт базы данных обратно в систему Protel, при экспорте в нее обязательно нужно включить соответствующие атрибуты положения. Без этих идентификаторов положения функция импортирования не сможет сопоставить поля базы данных с каждым экземпляром примитива на соответствующих листах.

Настройка области действия

После выбора примитивов и их атрибутов необходимо установить область действия функции экспортирования. Редактор схем имеет три варианта установки этого параметра: **Current Sheet** (текущий лист), **Current Project** (текущий проект) и **All Open Sheets** (все открытые листы).

Импорт из базы данных

Для вызова диалогового окна **Import from Database** (рис. 3.51) нужно воспользоваться командой меню **File » Import from Database**. Далее осуществляется выбор импортируемого типа примитива и затем с помощью кнопки **Browse** выбор файла базы данных. После настройки области действия (**Scope**) и операции обновления (**Action**) все готово для назначения полей базы данных (**Database Fields**) на атрибуты объектов (**Object's Attributes**).

Импортирование из базы данных — это процесс передачи информации из базы в поля атрибутов примитивов, находящихся в проекте. Для достижения этого необходимо

выполнить два действия: указать, что нужно извлечь из базы и определить куда должна поступить информация.

Первое действие выполняется с помощью назначения атрибутов схемных объектов на поля базы данных. Второе — с помощью определения области действия (**Scope**) и установки флагов ключевых атрибутов (**Key Attributes**).

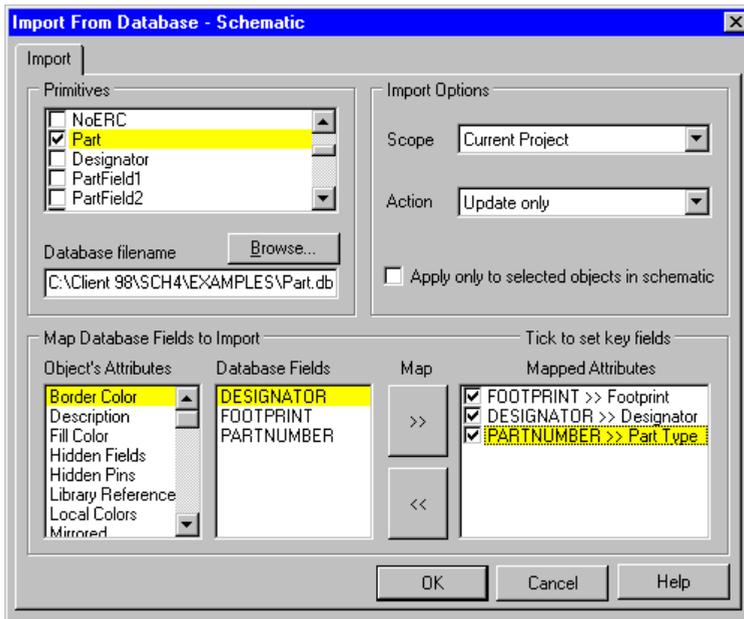


Рис. 3.51. Настройка импорта информации из базы данных.

Отображение атрибутов

В списке **Database Fields** выбирается поле, которое нужно импортировать из базы. Затем выбирается атрибут объекта схемы (**Schematic Object's Attributes**), который должен быть назначен на выбранное поле базы данных, и выполняется нажатие на кнопку >> (расположенную под меткой **Map**). Если содержимое выбранного поля может быть импортировано в выбранный атрибут, тогда они появятся в окне списка **Mapped Attributes**. В противном случае появится диалоговое окно, предупреждающее о том, что поле базы данных и атрибут объекта имеют различный тип.

Определение ключевых полей

Для того, чтобы определить критерий поиска объекта на листе, необходимо выбрать ключевое поле (или поля). Рядом с каждым атрибутом в списке **Mapped Attributes** находится элемент управления, который надо включить, если нужно, чтобы данный атрибут использовался в качестве ключевого поля.

Атрибуты, выбранные в качестве ключевых полей, должны однозначно идентифицировать экземпляр объекта. Для этих целей подходят атрибуты положения (**X** и **Y**), имя файла документа (**Document File Name**) и массив вершин. При определении ключевых полей нужно учитывать установленную область действия (**Scope**) функции импортирования. Если функция применяется к текущему листу (**Current Sheet**) достаточно атрибутов положения (или массива вершин). Если область действия больше чем один лист, то в список ключевых полей необходимо добавить имя файла документа (**Document File Name**).

Scope (область действия)

Также как и экспортирование, функция импортирования требует установку параметра области действия, имеющего три опции: текущий лист (**Current Sheet**), текущий проект (**Current Project**) и все открытые листы (**All Open Sheets**). Установка флажка **Only Selected Items** позволяет ограничить область действия функции, которая в этом случае будет распространяться только на выделенные объекты на схеме.

Import Options - Action (действие)

Существует три варианта использования импортируемой из базы данных информации. Информация может быть:

- Импортирована в качестве новых объектов (**imported as new objects**).
- Использована для обновления существующих объектов (**used to update existing objects**).
- Использована для обновления существующих объектов, если они существуют, иначе импортирована в качестве новых объектов (**update existing objects if they exist and add new objects if they do not**).

Обмен данными с другими системами проектирования

Редактор принципиальных схем имеет возможность обмениваться информацией с некоторыми системами автоматизированного проектирования, например, пакетом моделирования СВЧ устройств HP-EEsof, пакетом Xilinx FPGA, графическими редакторами принципиальных схем и печатных плат OrCAD DOS и OrCAD Capture.

Как правило, связь с этими пакетами осуществляется через список соединений, поэтому ниже приводятся различные способы создания списков соединений из редактора принципиальных схем.

В стандартный комплект поставки пакета Protel 99 SE включены примеры использования программ HP-EEsof и Xilinx, которые содержат документацию, описывающую методы обмена информацией с этими системами проектирования.

Преобразование информации из OrCAD

Система Protel 99 SE имеет прямой транслятор файлов проектов с расширением DSN, созданных в пакете OrCAD Capture 7.x и 9.x, в листы принципиальных схем и соответствующие библиотеки. Процесс преобразования полностью автоматизирован, то есть, исходный **DSN** файл просто импортируется в базу данных проекта, после чего формируются библиотеки элементов, а данные о схеме преобразуются в нужный формат.

Импорт и преобразование DSN файла

1. Создайте новую базу данных проекта.
2. Отключите все подключенные к системе библиотеки.
3. Щелкните правой кнопкой мыши на папке **Design Database**, и в появившемся всплывающем меню выберите команду **Import**.
4. В диалоговом окне **Import** выберите нужный **DSN** файл и нажмите кнопку **Open**.
5. Сначала **DSN** файл будет импортирован в базу данных проекта, а затем преобразован в формат системы Protel.

Система OrCAD не может хранить компоненты непосредственно в существующих листах принципиальных схем, и поэтому они содержатся в специальной внутренней библиотеке, называемой **Design Cache**.

Однако, нет необходимости сначала преобразовывать содержимое этой библиотеки в библиотеку Protel, а затем подключать ее к редактору схем, то есть, содержимое **Design Cache** автоматически преобразуется в нужную библиотеку при преобразовании листов принципиальных схем, которая оказывается подключенной к редактору схем.

Как отображается структура OrCAD Design

Все проекты импортируются с учетом изначальной (простой или сложной) иерархии. На рисунке 3.52 представлен вид проекта в системе OrCAD и результат импорта его в систему Protel.

Слева приведен вид проекта в системе OrCAD, справа — тот же проект, импортированный в систему Protel 99 SE. Иерархия проекта в Protel 99 SE создается непосредственно из иерархии проекта в OrCAD. Обратите внимание, что оригинал проекта **Simple.Dsn** остается храниться в новой базе данных проекта. Выполнив на нем двойной щелчок левой кнопкой мыши пользователь может повторно импортировать его.

При импорте проекта системы OrCAD:

- Каждая папка с принципиальными схемами преобразуется в лист схемы Protel с символами листов для каждой вложенной страницы схемы OrCAD внутри этой папки.
- Каждая страница схемы OrCAD преобразовывается в лист схемы Protel.
- Иерархические связи преобразуются во входы листа.
- Иерархические порты преобразуются в порты.

Связи между страницами схем OrCAD в папках схем OrCAD создаются при помощи межлистовых соединений. Преобразование такой горизонтальной структуры в принятую в Protel 99 SE вертикальную иерархическую структуру производится следующим способом:

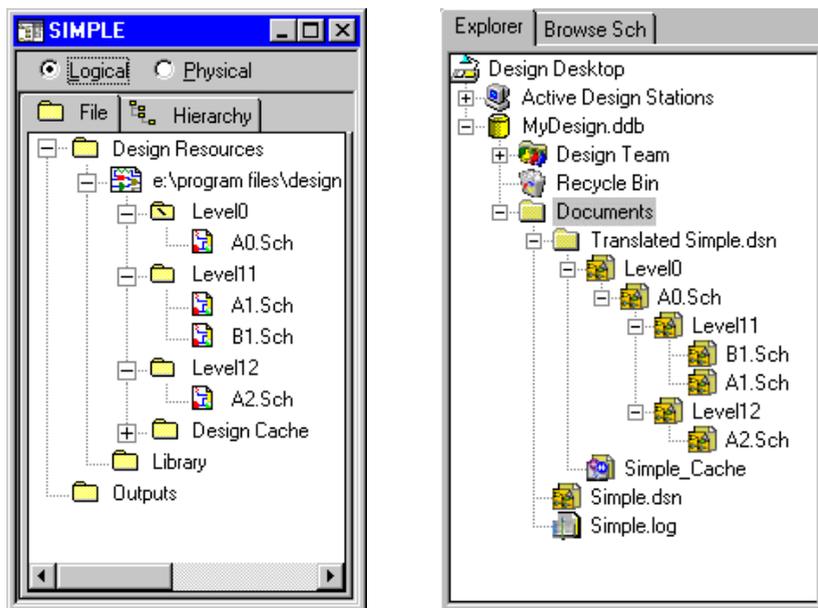


Рис. 3.52. Пример импорта в Protel 99 SE проекта OrCAD 9.2.

- Межлистовые соединения преобразовываются в порты и выделяются красным цветом.
- На символе листа, представляющего этот лист, создаются соответствующие входы листа, которые также выделяются красным цветом.
- Межлистовые соединения создаются на верхнем листе с помощью проводников и меток цепей.

Примечание

- Для корректной проверки правил электрических соединений (ERC) и синхронизации проекта платы необходимо использовать опцию **Sheet Symbol/Port Connection Net Identifier Scope**.
- Расположение некоторых пользовательских символов может быть повторено не совсем точно, поэтому рекомендуется проверять расположение таких символов сразу после преобразования.
- Содержащиеся в схемах картинки не импортируются, вместо них появятся красные прямоугольники.

- Шинные выводы в системе Protel 99 SE не поддерживаются, поэтому они преобразовываются в обычные выводы, о чем программа выдает соответствующее предупреждение.
- В системе OrCAD возможно соединение шин с разными именами, что в Protel 99 SE недопустимо.

Обмен информацией с системой P-CAD 2000

Редактор схем Protel 99 SE имеет встроенный двунаправленный транслятор в формат P-CAD 2000 ASCII. Процесс преобразования полностью автоматизирован, то есть, исходный SCH файл просто импортируется в базу данных проекта, после чего формируются библиотеки элементов, а данные о схеме преобразуются в нужный формат.

1. Создайте новую базу данных проекта.
2. В папке **Documents** создайте новый лист принципиальной схемы и откройте его.
3. Выполните команду меню редактора схем **File » Import » PCAD 2000 ASCII**, укажите импортируемый файл и нажмите кнопку **OK**.

Результат импорта стандартного примера P-CAD 2000 **Digdemo.sch**, представляющего собой проект из двух листов принципиальных схем, показан на рисунке 3.53.

- Каждый проект P-CAD 2000 преобразуется в лист схемы Protel с расширением **.PRJ** с символами листов для каждого листа схемы P-CAD.

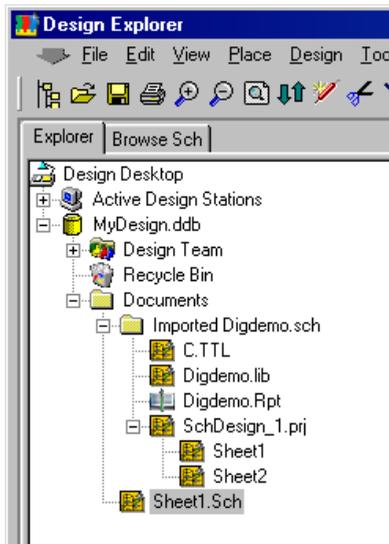


Рис. 3.53. Результат импорта в Protel 99 SE двухлистового проекта P-CAD 2000.

- Каждый лист схемы P-CAD 2000 преобразовывается в лист схемы Protel.
- Иерархические связи преобразуются во входы листа.
- Иерархические порты преобразуются в порты.
- Если на схемах в P-CAD использовались рамки, то будет создан отдельный файл схемы с расширением .TTL, в котором они будут нарисованы разными цветами.
- Также автоматически будет создана библиотека, содержащая все используемые в проекте компоненты.
- Первоначально созданный пустой лист схемы остается пустым и его можно удалить.

Связи между листами схем P-CAD создаются при помощи портов, горизонтальная структура которых преобразуется в принятую в Protel 99 SE вертикальную иерархическую структуру. Межлистовые соединения создаются на верхнем листе с помощью проводников и меток цепей.

При экспорте проектов из Protel 99 SE в формат SCH P-CAD 2000 ASCII, следует помнить, что P-CAD не может открывать ASCII файлы, в которых присутствует маленькая русская буква "я". Рекомендуется открыть SCH ASCII файл с помощью любого текстового редактора и автоматически заменить все маленькие "я" на большие "Я" или просто удалить их.

Так как в версии P-CAD 2001 в формат SCH ASCII не было внесено изменений, под-

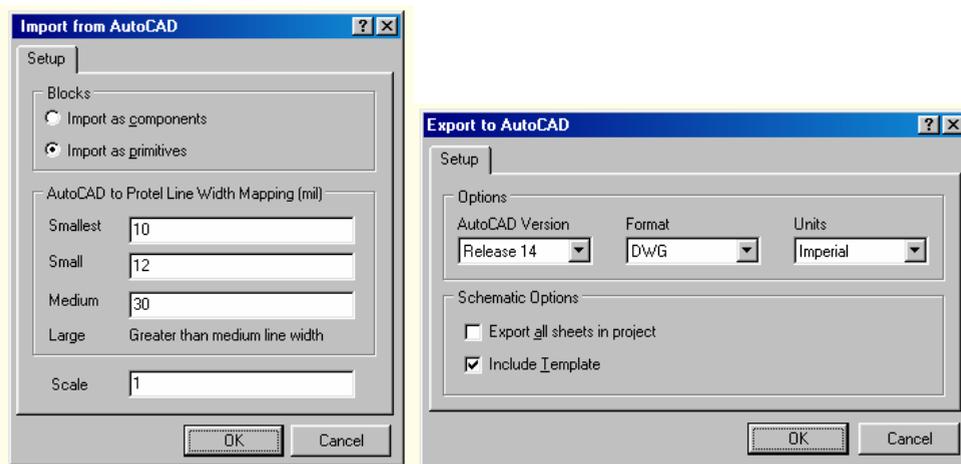


Рис. 3.54. Окна настройки импорта и экспорта файлов системы AutoCAD.

держивается полная двунаправленная передача принципиальных схем и с этой версией. Обмен данными с ранними версиями P-CAD Master Designer 4.5 – 8.7 на уровне принципиальных схем без использования специальных конвертеров невозможен.

Интерфейс с механическими САПР

Система Protel 99 SE имеет двунаправленный интерфейс связи с механическими САПР через файлы форматов **DWG** и **DXF**. Для запуска процессов импорта или экспорта используются команды меню редактора схем **File » Import** или **File » Export** соответственно (рис. 3.54).

Создание списков соединений

Списки соединений используются в большинстве систем САПР. Они представляют собой особым образом выполненный перечень всех входящих в схему компонентов и их соединений.

Как правило, список соединений представляет собой файл в формате ASCII. Формат списка соединений включает в себя описание компонентов, в которое входит позиционное обозначение и тип корпуса, и межвыводных соединений, определяющих все цепи. Автоматическая загрузка топологического посадочного места на печатную плату позволяет избежать многих ошибок, присущих процессу проектирования.

Форматы списка соединений

Список соединений может быть представлен в разных форматах, но обычно это ASCII файл, содержащий следующую информацию:

- Описание компонентов схемы.
- Список всех межвыводных соединений в схеме.
- Некоторые форматы списка соединений могут включать дополнительную информацию из текстовых полей компонентов или цепей. Эта информация может быть использована для правильной передачи информации в программу моделирования или редактор печатных плат, в случаях когда стандартных иерархических методов обеспечения связности не достаточно. В стандартный комплект поставки включены несколько примеров таких проектов.

Некоторые форматы списков соединений объединяют в одном описании данные об элементах и их соединениях. Другие, в том числе форматы Protel 99 SE, разделяют эти данные на две отдельные секции.

Будучи обычными текстовыми файлами, списки соединений могут быть без особого труда преобразованы в другие форматы при помощи простой написанной пользователем программы. Кроме того, списки соединений могут быть созданы или отредактированы при помощи привычного обычного текстового редактора.

Если список соединений вводится вручную, необходимо проследить за тем, чтобы он был сохранен в неформатированном виде, то есть в обычной текстовой форме, так как скрытые управляющие символы могут сделать список нечитаемым.

Формирование списка соединений

Список соединений может быть сформирован в любое момент работы над проектом. Для этого надо выполнить команду меню **Design » Create Netlist** (горячие клавиши: **T, N**), после чего откроется диалоговое окно **Netlist Creation**.

Выходной формат списка соединений (Netlist Output Format Options)

Ниже приводится перечень поддерживаемых выходных форматов списков соединений.

Algorex	EEsof Touchstone	Protel – Hierarchical
AppliconBRAVO	FutruceNet	Protel Wirelist
AppliconLEAP	Hilo	Racal Redac
Cadnetix	Integraph	Scicards
Calay	Mentor BoardStation 6	Spice
Calay90	Multiwire	Spice Hierarchical
Case	OrCAD – PLDnet	Star Semiconductor
CBDS	OrCAD - PCB II	Tango
ComputerVision	PADS Ascii	Telesis
EDIF 2.0	PCAD	Vectron
EDIF 2.0 Hierarchical	PCAD NLT	VHDL
EEDesigner	Protel	Xilinx XNF
EEsof Libra	Protel 2	

Область действия идентификаторов цепей (Net Identifier Scope)

Область действия идентификаторов цепей определяет способ задания межлистовых соединений. Существует два способа создания межлистовых соединений: вертикальный (из входов листа вниз к соответствующим портам) и горизонтальный (непосредственно из порта или метки цепи в соответствующий порт или метку цепи). Очень важно, что область действия идентификаторов цепей устанавливается в зависимости от существующей структуры проекта.

Для получения более подробной информации о структуре многостраничного проекта обратитесь к разделу *Управление многострочными и иерархическими проектами*.

Существует три варианта задания области действия идентификаторов цепей:

Net Labels and Ports Global (глобальные метки цепей и порты)

Эта опция подразумевает, что используемые метки цепей относятся ко всем листам текущего проекта. Другими словами, цепи являются глобальными и каждый экземпляр метки цепи и порта оказывается соединенным со всеми одноименными меткам цепей и портам (заметим, что порты не присоединяются к меткам цепей, а метки не присоединяются к портам). Эта модель работает аналогично системе Protel Schematic 3 (DOS), где метки цепей всегда глобальны во всех листах проекта. В этом варианте создается горизонтальные межлистовые соединения.

Only Ports Global (глобальные только порты)

При установке этой опции метки цепей трактуются как локальные, то есть связанные только в пределах каждого отдельного листа. Межлистовые соединения осуществляются посредством идентично помеченных портов. Эта модель работает аналогично плоской модели проекта OrCAD SDT. В этом варианте также создаются только горизонтальные межлистовые соединения.

Sheet Symbols / Port Connections (соединения символы листа/порты)

В этом варианте межлистовые соединения цепей выполняются только с помощью входов символов листа и портов на подчиненных листах. Полагается, что порты будут соединяться только с идентично поименованными входами листа в их символах листа на родительских листах. Эта модель работает аналогично иерархической модели проекта OrCAD SDT. В этом случае создаются только вертикальные межлистовые соединения.

Режим преобразования (Sheets to Netlist)

Active Sheet (активный лист) — составить список соединений только для активного листа.

Active Project (активный проект) — открыть все листы, входящие в иерархический проект, и составить для них единый список соединений.

Active Sheet plus sub-sheets (активный лист и все подчиненные листы) — открыть все листы, лежащие ниже активного листа в иерархическом проекте, и составить для них единый список соединений. Этот вариант используется если текущий проект является частью большего проекта.

Другие опции (Options)

Имеется ряд других опций, определяющих содержание списка соединений:

Добавлять номера листов к именам локальных цепей (Append Sheet Number to Local Net Names)

Выполняет добавление номера листа, задаваемого на вкладке **Organization** в диалоговом окне **Design Option**, ко всем цепям. Если был выбран такой вариант области действия идентификаторов цепей, при котором метки цепи локальны, то эта опция добавит номер листа к цепи, обеспечивая тем самым уникальность каждой цепи в списке соединений. Эта опция может быть также использована при отладке проблем, связанных с неправильным соединением цепей на разных листах при составлении списка соединений. При добавлении номера листа можно без труда определить, на каком листе находится не присоединенная цепь.

Использовать компоненты в качестве символов листа (Descend Into Sheet Parts)

Эта опция включается, когда некоторые компоненты на схемах используются как иерархические символы листов. Для того, чтобы система восприняла элемент схемы как символ листа, необходимо в поле **Sheet Path** диалогового окна **Edit Part** указать имя подчиненного листа. Список соединений в этом случае будет содержать информацию о всех листах, лежащих ниже компонентов, используемых в качестве символов листов. Для получения дополнительной информации обратитесь к разделу *Управление многолистовыми и иерархическими проектами*.

Форматы списков соединений системы Protel 99 SE

Если список соединений планируется загружать в редактор печатных плат, то длина позиционных обозначений и имен топологических посадочных мест не должна превышать 12 символов. Длина комментариев может достигать 32 символов. Имена цепей могут содержать до 20 символов. Длина номеров выводов в списке соединений не должна превышать 4 символа. Отметим, что использование пробелов здесь недопустимо.

Количество компонентов и узлов, которые могут быть включены в список соединений формата **Protel** или **Protel2**, ограничено только ресурсами памяти используемого компьютера.

Формат списков соединений Protel

Стандартный формат списка соединений Protel представляет собой это простой текстовый файл в формате ASCII, разделенный на две части. В первой части приведено описание всех компонентов:

[Начальный разделитель компонентов.
U8	Позиционное обозначение компонента.
DIP16	Имя топологического посадочного места. Посадочное место с этим именем должно быть доступно в подключенных библиотеках компонентов печатных плат.
74LS138	Тип элемента (или комментарий).
(пусто)	Эти три строки не используются.
(пусто)	
(пусто)	
]	Конечный разделитель компонентов.

Вторая часть списка соединений формата Protel описывает все цепи:

(Начальный разделитель цепей.
CLK	Имя цепи.
U8-3	Первый компонент (по позиционному обозначению) и номер вывода. Номера и количество выводов топологических посадочных мест должны точно соответствовать.
J21-1	Второй узел цепи.
U5-5	Третий узел.
)	Конечный разделитель цепей.

Формат списков соединений Protel 2.0

Этот формат похож на стандартный список соединений Protel, но здесь добавлены несколько текстовых полей, содержащих данные, необходимые для моделирования, и вспомогательную информацию, а также топологические директивы, содержащие информацию для трассировщика. Такой формат может загружать расширенный редактор плат версии 2.0 и выше.

PROTEL NETLIST 2.0	Заголовок списка соединений.
[Начальный разделитель компонентов.
DESIGNATOR	(Сначала идет имя поля)
U1	Позиционное обозначение компонента.
FOOTPRINT	

DIP20	Топологическое посадочное место.
PARTTYPE	
AmPAL16L8	Тип элемента.
DESCRIPTION	
Description	Поле описания из принципиальной схемы.
PART FIELD 1	(Имя поля может быть изменено средствами редактора принципиальных схем)
Part Field 1	Поля элементов (1—16) из принципиальной схемы.
...	
LIBRARYFIELD1	
Library Part Field 1	Библиотечные поля (1—8) из библиотеки элементов принципиальных схем.
...	
]	Конечный разделитель компонентов.
(Начальный разделитель цепей.
VCC	Имя цепи.
U1-20 AMPAL16L8-VCC POWER	Первый узел в цепи. Включает: обозначение вывода компонента. (пробел) Имя вывода в компоненте. (пробел) Электрический тип вывода.
U2-14 4001-VCC POWER	Последний узел в цепи.
)	Конечный разделитель цепей.
{	Начальный разделитель топологических директив.
TRACK	(Сначала идет имя поля).
10	Ширина проводника (в тысячных долях дюйма — милах).
VIA	
50	Диаметр переходных отверстий (в милах).
NET TOPOLOGY	

SHORTEST	Топология цепи для трассировки.
ROUTING PRIORITY	
MEDIUM	Приоритет трассировки.
LAYER	
UNDEFINED	Слой для трассировки.
}	Конечный разделитель топологических директив.

Создание списка соединений в формате VHDL

Программа формирования списка соединений в формате VHDL базируется на стандарте IEEE VHDL Standard 1076/93, определяющим стандартные логические векторы согласно IEEE std_logic 1164. Проект может быть смешанным, то есть содержать как листы принципиальных схем, так и код VHDL.

Библиотеки элементов схем

Для хранения элементов схем используются библиотеки микросхем серий 2000 — 7000 производства фирмы Xilinx, находящиеся в базе данных **Xilinx.Ddb**.

Создание смешанного проекта

Верхним листом смешанного проекта должен быть лист принципиальной схемы. Ссылка из символа листа на исходный VHDL файл производится так же, как и на лист схемы, то есть посредством ввода имени файла в поле **Sheet Symbol Filename**.

На верхний лист VHDL проекта также может быть сделана ссылка из физического компонента, расположенного на чертеже печатной платы. Для этого нужно ввести имя верхнего листа VHDL в поле **Path Field** требуемого физического компонента.

Как в проекте формируется список соединений

Для верхнего листа в проекте создается объявление объекта (**Entity declaration**), а схема на этом листе транслируется в соответствующее ей архитектурное объявление (**Architecture declaration**). Каждый компонент и символ листа на этом листе становится VHDL компонентом в пределах этого архитектурного объявления.

Этот процесс затем продолжается для каждой пары символ листа/дочерний лист на верхнем листе — для символа листа создается объявление объекта, а компоненты и символы листа на этом листе становятся VHDL компонентами в пределах соответствующего архитектурного объявления.

Создание списка соединений проекта

Для создания списка соединений необходимо выполнить команду **Design » Create Netlist** из меню редактора принципиальных схем. Затем в диалоговом окне **Netlist Creation** нужно установить параметр выходного формата (**Output Format**) в значение VHDL. Область действия команды задается тремя способами:

- **Active Sheet** (активный лист) — составить список соединений только для активного листа.

- **Active Project** (активный проект) — открыть все листы, входящие в иерархический проект, и составить для них единый список соединений.
- **Active Sheet plus sub-sheets** (активный лист и все подчиненные листы) — открыть все листы, лежащие ниже активного листа в иерархическом проекте, и составить для них единый список соединений. Этот вариант используется если текущий проект является частью большего проекта.

Некоторые рекомендации по разработке схем по ГОСТ

Редактор принципиальных схем системы Protel 99 SE имеет ряд особенностей, которые необходимо учитывать при оформлении схем согласно требованиям ЕСКД. В общем случае система Protel 99 SE не предназначена для оформления схем и чертежей по ГОСТ, однако ее гибкость позволяет решить большинство проблем.

Настройки сеток

Главной проблемой редактора Advanced Schematic является то, что он предполагает использование только одной системы измерения — дюймовой. Это поначалу, как правило, пугает пользователей, имеющих навыки работы в программах ACCEL EDA и P-CAD, где имеется возможность использования как дюймовой, так и метрической системы мер.

Однако, следует учитывать, что в отличие от редактора чертежей печатных плат, где все графические примитивы должны иметь истинные геометрические размеры, оформление схем может вестись в дискретах — неких условных единицах измерения. Это связано с тем, что при создании схем разработчики пользуются готовыми условными графическими изображениями из соответствующих библиотек. Главное, чтобы выведенная на печать схема имела правильные размеры в миллиметрах, чего легко добиться подбором масштаба печати.

Редактор принципиальных схем системы Protel имеет дискретность 0.01 дюйма, что составляет около 0.25 мм. Таким образом, процесс создания схемы при правильном назначении размеров сеток не будет ничем отличаться от прорисовки схемы на миллиметровой чертежной бумаге. Редактор принципиальных схем имеет три типа сетки: видимая **Visible Grid**, облегчающая ориентирование по схеме; шаг **Snap Grid**, определяющий дискретность перемещения курсора по схеме; и электрическая сетка **Electrical Grid**, задающая некоторую окрестность (в дискретах) вокруг концов электрических объектов, при попадании в которую курсора на экране отображается "горячая точка", облегчающая соединение электрических объектов.

Значения всех этих сеток устанавливаются на вкладке **Sheet Options** диалогового окна **Document Options**, вызываемого командой меню **Design » Options**. Так как для оформления схем согласно требованиям ЕСКД достаточно точности 0.5 мм, рекомендуется рисовать схемы в уменьшенном масштабе, что дает возможность, во-первых, без проблем использовать УГО микросхем из фирменных библиотек, имеющих шаг ме-

жду выводами 10 дискретов, а во-вторых, почти вдвое увеличить максимальный размер листа схемы.

Для получения оптимального масштаба при выводе схемы на печать рекомендуется установить шаг сетки **Snap Grid** равный 2 дискрета, в этом случае шаг перемещения курсора будет составлять около 1 мм. Шаг видимой сетки необходимо задать равным 10 дискретов, что соответствует 5 мм. При таком выборе шага размер электрической сетки должен несколько превышать его, иначе "горячая точка" будет появляться только при попадании в конец электрического объекта. По ГОСТ проводники должны располагаться на расстоянии не ближе 5 мм друг к другу, а значит, значение электрической сетки должно лежать в пределах от 2 до 10 дискретов. Рекомендуется оставить задаваемое по умолчанию значение 8 дискретов.

При таких установках для правильного вывода схем на печать необходимо задавать масштаб равный 196.8%, что соответствует удвоенному масштабу так называемого "русского дюйма", равного ровно 25 мм (вместо 25.4 мм). Исходя из этого значения масштаба, можно определить максимальный размер листа принципиальной схемы, который можно создать в редакторе Advanced Schematic. Максимальный размер схемы равен 64 дюйма или 6400 дискретов, что в масштабе 196.8% составляет 3200 мм и превышает все стандартные форматы схем по ГОСТ. При задании пользовательского размера листа схемы необходимо обязательно вводить число зон по осям X и Y не равное нулю, в противном случае, не будет выполняться операция автоматической расстановки позиционных обозначений (команда **Tools** » **Annotate**).

Отметим также, что в редакторе принципиальных схем системы Protel отсутствует возможность точного задания толщины линии (в отличие от системы P-CAD). Толщина линий здесь определяется атрибутом **Width**, значение которого выбирается из выпадающего списка, имеющего четыре опции: **Smallest** (очень тонкая), **Small** (тонкая), **Medium** (средняя), **Large** (толстая). Согласно требованиям ГОСТ толщина проводников на принципиальных схемах должна быть равна 0.5 — 1 мм, и при масштабе печати 196.8% этому значению соответствует толщина Small. Это же значение необходимо использовать для прорисовки основных линий, а для вспомогательных тонких — значение **Smallest**.

Так как в комплекте с системой Protel 99 SE поставляется набор шаблонов стандартных форматов, нет необходимости разрабатывать их заново, достаточно лишь правильно задать шаблон листа. Используемый по умолчанию шаблон листа задается на вкладке Schematic диалогового окна **Preferences (Tools** » **Preferences**). В случае, когда разрабатываемая схема перестает помещаться на выбранном ранее шаблоне листа, его можно без труда изменить, сначала удалив существующий, а потом назначив новый шаблон с помощью соответствующих команд меню **Design** » **Template**.

Использование шаблонов рекомендуется также из-за того, что все составляющие их графические примитивы не будут доступны для редактирования при разработке схемы, а значит, нельзя случайно повредить, например, стандартный вид основной надписи. Кроме того, шаблон содержит ряд специальных текстовых строк, которые будут автоматически заполняться при вводе соответствующей информации на вкладку **Organization** диалогового окна **Document Options**.

Разработка библиотечных элементов

Главное внимание следует уделить разработке условных графических обозначений элементов. В редакторе библиотек также имеется ряд особенностей, которые необходимо учитывать при создании УГО. Прежде всего, здесь должны быть установлены размеры сеток, аналогично тому, как это делается в редакторе схем. Как правило, точность прорисовки графических примитивов, составляющих УГО элементов, не

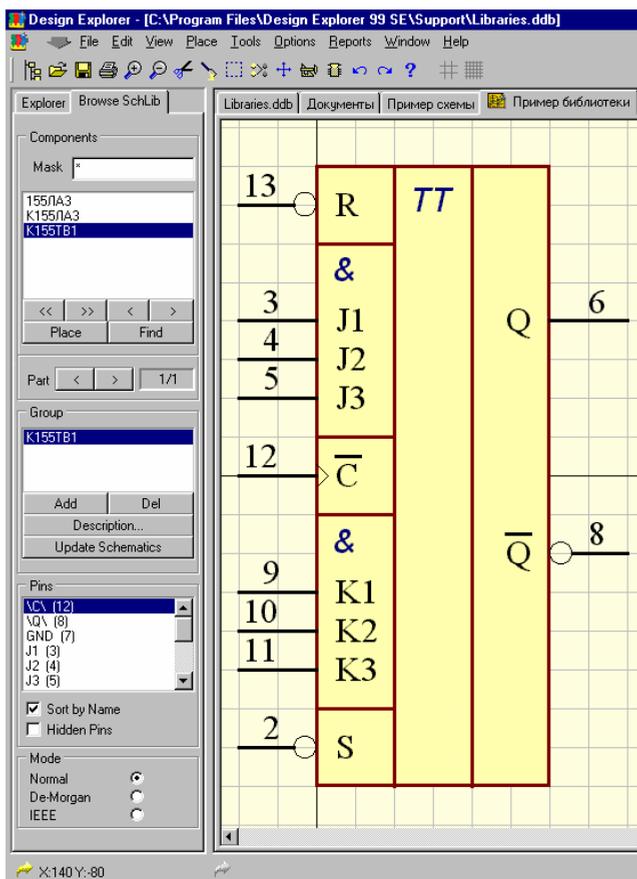


Рис. 3.55. Создание библиотечного элемента по ГОСТ.

превышает 0.5 мм, что соответствует шагу сетки **Snap Grid** равному 1 дискрету. Видимая сетка должна быть равна 10 дискретов или 5 мм.

Для удобства рисования принципиальных схем больших размеров рекомендуется при создании УГО размещать электрические концы выводов в узлы сетки с шагом 5 мм, а один из выводов разместить в точке начала координат. В этом случае концы всех

выводов будет удобно привязывать к узлам видимой сетки. При размещении на схеме в качестве точки привязки будет автоматически выбран один из выводов элемента, а все другие выводы впишутся в сетку 5 мм.

Большинство УГО различных модификаций дискретных элементов: диодов, транзисторов, переключателей, резисторов и конденсаторов поставляется лицензионным пользователям в комплекте с программой. В связи с большим числом наименований, разработка УГО цифровых и аналоговых микросхем затруднена, поэтому ограничимся лишь рекомендациями по их созданию.

Прежде всего, настоятельно не рекомендуется перекачивать библиотеки из старых версий системы P-CAD. Эффект от использования "старых наработок" будет минимальным, так как в любых, даже самых хороших библиотеках обязательно будут присутствовать ошибки, поиск и исправление которых займет гораздо большее время, чем создание компонента заново. В стандартный комплект поставки системы Protel 99 SE входит набор библиотек компонентов, сертифицированный по стандарту ISO 9001. Компания постоянно обновляет библиотеки и дополняет их новыми компонентами согласно информации, поступающей от фирм производителей электронных компонентов. Все библиотеки можно беспрепятственно скачать с сайта компании Protel.

Сейчас большинство разработчиков использует в своих проектах импортные компоненты, поэтому достаточно выбрать нужный компонент из фирменной библиотеки, изменить его УГО согласно ГОСТ и сохранить в собственной пользовательской библиотеке. При таком подходе в пользовательскую библиотеку будет перенесена различная служебная информация, например, тип топологического посадочного места, описание, параметры SPICE модели, число ошибок будет минимальным. Для отечественных микросхем, имеющих зарубежный прототип, рекомендуется по справочной документации определить его, найти в стандартной библиотеке, изменить, переименовать и сохранить в пользовательской библиотеке под новым именем. Если отечественная микросхема не имеет зарубежного аналога, рекомендуется создать компонент заново, основываясь на паспортных данных устройства.

При создании УГО компонента следует учитывать, что толщина линий, которыми будет прорисован компонент, задается в редакторе библиотек. В редакторе схем можно будет изменить только толщину проводников. Как было сказано выше, при прорисовке УГО рекомендуется использовать линии со значением **Small** атрибута **Width**, что при масштабе 196.8% соответствует толщине линии 0.5 мм. Выбор такого масштаба значительно упрощает работу со стандартными выводами, которые имеют стандартную и неизменяемую ширину линии **Small**.

Длину выводов и шаг между выводами необходимо выбирать кратной 5 мм или 10 дискретам. Все атрибуты вывода задаются на вкладке **Properties** (рис. 3.56). Длина вывода здесь задается в поле **Pin Length**, имя вывода — в поле **Name**, номер вывода — в поле **Number**.

Ориентация вывода задается в выпадающем списке **Orientation**. Угол поворота 0° соответствует такому расположению вывода, когда его электрический конец (конец, где появляется "горячая точка") находится справа. При изменении длины вывода точкой привязки считается неэлектрический конец.

Номер вывода располагается над точкой привязки, а имя вывода — слева от нее. Горизонтальное смещение номера вправо от точки привязки задается положительным числом дискретов в поле **Pin Number Margin**, а имени вывода влево — положительным числом в поле **Pin Name Margin** на вкладке **Schematic** диалогового окна **Properties (Options » Properties)**. При необходимости здесь допускается использование отрицательных чисел. Шрифт отображения номера и имени выводов является системным и задается в редакторе схем. То, что надписи выполнены системным шрифтом, означает, что при повороте элемента на схеме на 180° надписи сохраняют ориентацию. Тем не менее, допускается добавления к УГО произвольных текстовых надписей, которые в редакторе схем будут считаться составляющими частями УГО и будут поворачиваться синхронно с элементом. Шрифт и цвет таких надписей может быть выбран произвольным.

Правильный выбор электрического типа вывода в поле **Electrical Type** позволит корректно выполнять проверку правил электрических соединений на схеме. При использовании символов инверсии (**Dot Symbol**) и синхронизации (**Clk Symbol**) рекомендуется прорисовать поверх них соответствующий толстый значок. Для отображения имени и номера вывода на схеме в атрибутах вывода в редакторе библиотек необходимо включить опции **Show Name** и **Show Number** соответственно. Для выводов питания и заземления цифровых микросхем рекомендуется включать опцию **Hidden**, что позволит не отображать эти выводы на принципиальной схеме. Все скрытые выводы, имеющие одинаковое имя, при генерации списка соединений или проверке электрических правил будут связаны одноименной цепью. Для аналоговых компонентов выводы питания и заземления скрывать не рекомендуется.

Редактор элементов схем системы Protel позволяет присваивать электрическим объектам инверсные обозначения. В P-CAD 2000 это делается с помощью символа "~" —

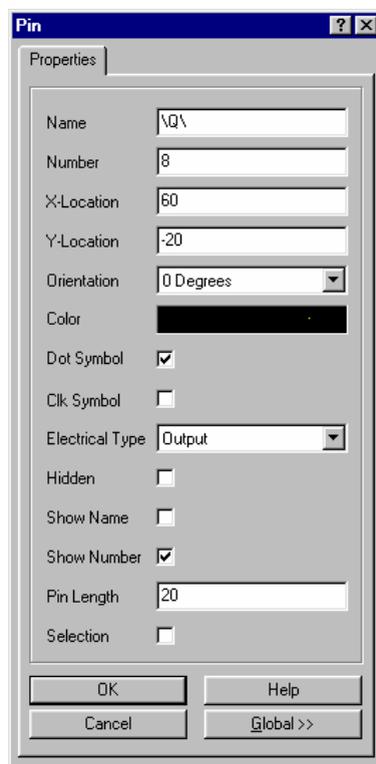


Рис. 3.56. Задание параметров вывода элемента.

все, что написано после этого символа будет иметь значок инверсии. В Protel инверсия назначается символом "\" и имеет два режима: обычный, когда в надписи инвертируется символ, предшествующий значку "\" (на вкладке **Graphical Editing** окна **Preferences** выключена опция **Single \" Negation**); и универсальный, когда значок "\" ставится в начале надписи, и инвертируются все символы надписи (опция **Single \" Negation** включена). Аналогичная опция имеется в редакторе схем (рис. 3.57). Первый способ позволяет чередовать в имени объекта инверсные и обычные символы, однако в этом случае символы "\" будут отображаться в списке **Pins** и затруднять считывание имен выводов.

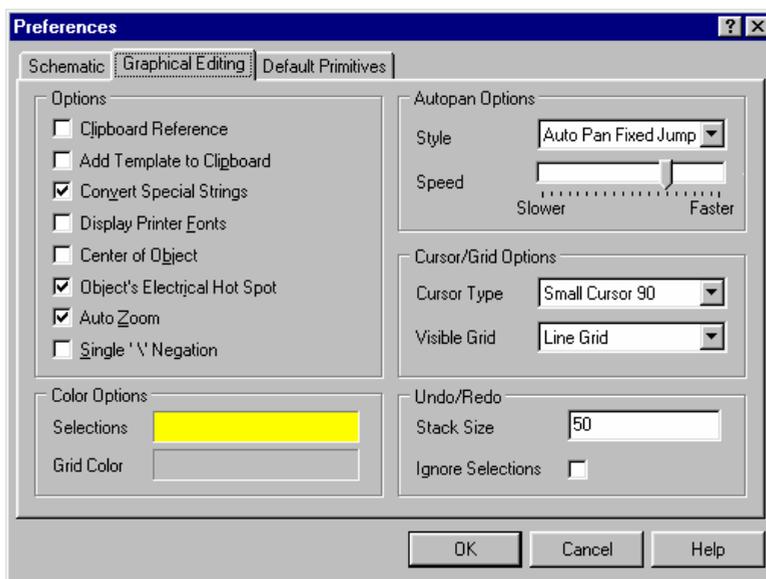


Рис. 3.57. Настройка параметров графического редактирования.

У размещенного на схеме элемента будут автоматически отображаться поля **Part Type** (имя компонента в библиотеке) и **Part Designator** (позиционное обозначение). При повороте компонента эти текстовые надписи ведут себя как системные (то есть при повороте на 180° возвращаются в исходное состояние), но допускается их индивидуальный поворот и перемещение независимо от компонента, а также редактирование атрибутов (локальное и глобальное). При включении опций **Hidden Fields** и **Field Names**, на схеме будут показаны содержимое 16-и пользовательских полей и их названия. Неизменяемые библиотечные текстовые (**Read-Only Fields**) поля компонентов на схему не выводятся и служат для передачи из библиотеки в схему различной служебной информации, например, параметров SPICE или SimCode модели. Все текстовые поля компонента можно перемещать, поворачивать и редактировать индивидуально.

Пользовательские поля **Part Fields** заполняются в редакторе схем. В редакторе библиотек УГО можно лишь переименовать названия этих полей. Делается это на вкладке **Part Field Names** диалогового окна **Component Text Fields**, как показано на рисунке 3.58 (**Tools** » **Description**). В дальнейшем уникальные имена полей могут быть использованы в качестве ключевых для связывания проекта с внешними базами компонентов. Этот механизм позволяет избежать рутинной операции заполнения текстовых полей для перечня используемых материалов (**Bill of Materials**) и оперативно обновлять информацию о компонентах за счет "горячей связи" с внешней базой данных. Не рекомендуется использовать для этой цели неизменяемые библиотечные поля, так как большая часть из них зарезервирована под параметры для программы моделирования.

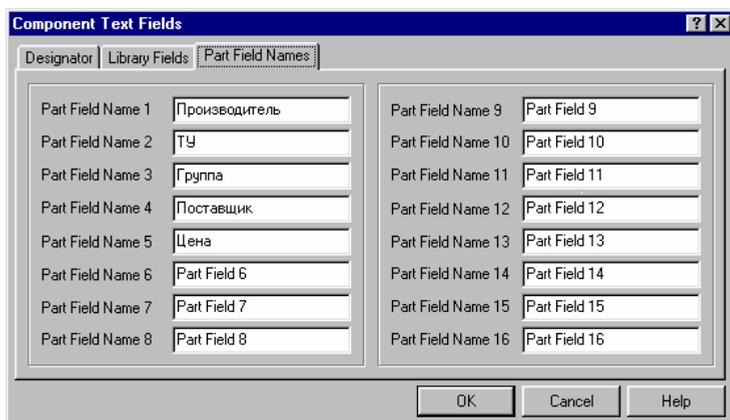


Рис. 3.58. Задание заголовков пользовательских полей.

Прорисовка портов питания

Отдельно следует остановиться на способах прорисовки символов заземления и питания. Редактор принципиальных схем системы Protel использует для этого встроенные символы портов питания (**Power Port**). Для размещения символа вывода питания следует выполнить команду **Place** » **Power Port** и войти в режим редактирования атрибутов символа. В открывшемся окне необходимо задать в поле **Style** стиль порта **Circle**, а в поле **Net** имя цепи, например **VCC**. Если в схеме присутствуют цифровые компоненты, выводы питания которых, как правило, являются скрытыми, то необходимо задать такое имя порта питания, которое будет точно совпадать с именем соответствующих скрытых выводов (регистр роли не играет). В этом случае при генерации списка соединений система автоматически распознает все цепи и правильно создаст список соединений.

Иногда требуется изобразить вывод питания не в виде круга, а в виде стрелки. В этом случае следует изменить стиль порта на **Arrow**. Больше проблем возникает при прори-

совке портов заземления. Графика этих символов также задается в поле **Style**, но она значительно отличается от описанной в ГОСТ. Наиболее близкое к требованиям ГОСТ начертание имеет стиль **Power Ground**, который по умолчанию подразумевает имя **GND**. Можно попробовать использовать стиль **Bar**, но в этом случае рядом с символом всегда будет отображаться имя цепи.

Если встроенные начертания символов портов питания пользователя не устраивают, то рекомендуется использовать специально разработанные для этого УГО, которые компания ЭлектронТрейд предоставляет своим лицензионным пользователям. На рисунке 3.59 изображены различные способы прорисовки символов питания и заземления. Проблем не возникает, если требуется просто нарисовать принципиальную схему и вывести на печать.

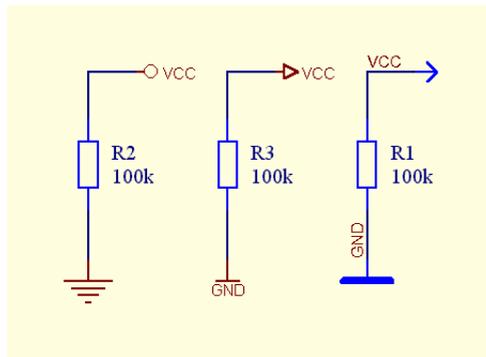


Рис. 3.59. Различные способы прорисовки символов питания и заземления.

Если же для разработанной принципиальной схемы требуется спроектировать печатную плату (а именно это и есть главное назначение системы Protel), то необходимо выполнить ряд операций, которые позволят правильно сгенерировать список соединений. Во-первых, всем цепям, соединенным с пользовательскими символами питания и заземления необходимо присвоить имя, разместив рядом с проводником метку цепи (**Net Label**). Чтобы метки цепей не выводились на печать, их необходимо выделить с помощью операции глобального редактирования, удалить нажатием комбинации клавиш **CTRL+DELETE**. Далее следует настроить и выполнить печать, после чего вернуть схему в исходное состояние двумя последовательными запусками команды **Edit » Undo**. Для быстрой реализации такой последовательности действий рекомендуется написать соответствующий макрос на языке Client Basic. Во-вторых, любой библиотечный символ вывода питания или заземления будет занесен в список соединений, а значит при загрузке его в редактор печатных плат из списка в окне **Load/Forward Annotate Netlist** следует исключить макросы, добавляющие эти компоненты на плату, а также их выводы к соответствующим цепям.

Более изощренный способ подготовки схемы к созданию печатной платы, заключается в написании макроса, который будет заменять все пользовательские символы питания и заземления на встроенные символы портов питания, генерировать корректный список соединений, а затем возвращать схему в исходное состояние. Этот способ может показаться слишком сложным, но именно он позволяет полностью решить проблемы несоответствия требований оформления схем по ГОСТ и методики проектирования в системе Protel.

Особенности разработки иерархических проектов

Гораздо больше проблем из-за этих несоответствий возникает при построении иерархических проектов.

Российские стандарты позволяют строить сколь угодно сложные иерархические проекты, однако их оформление резко отличается от принятого в западных системах проектирования. Иерархические вертикальные и горизонтальные связи между схемами здесь описываются разъемами по принципу контакт-адрес. В системе Protel эти связи описываются через порты (**Port**), символы листа (**Sheet Symbol**) и входы листа (**Sheet Entry**). Графика этих символов не может изменена настолько, чтобы полностью соответствовать требованиям ГОСТ.

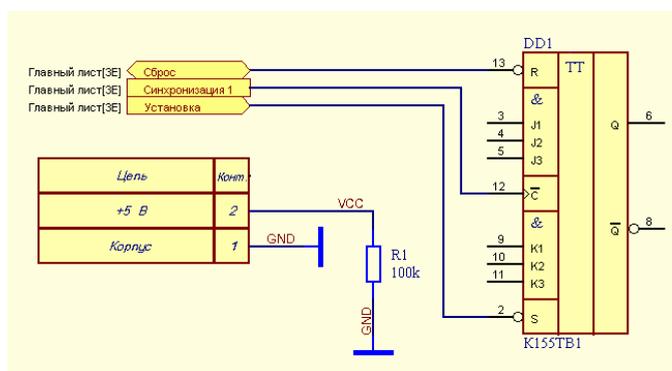


Рис. 3.60. Различные способы прорисовки портов.

На рисунке 3.60 изображен фрагмент принципиальной схемы подчиненного листа, где часть портов (слева сверху) цепей изображена с помощью встроенных объектов, а другая часть (слева внизу) — с помощью пользовательских символов. На вышестоящем уровне иерархии эта схема будет представлена символом листа, приведенным на рисунке 3.61. Применение пользовательских символов дает возможность описывать иерархические связи только посредством глобальных меток цепей, что соответствует "плоской" иерархии (Модель 2) и значительно ограничивает функциональность редактора схем системы Protel.

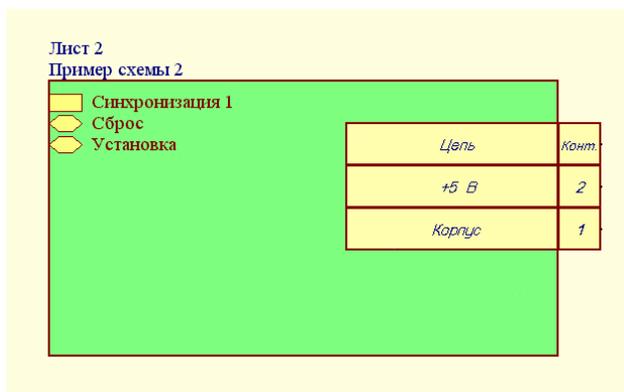


Рис. 3.61. Различные способы прорисовки символа листа.

В общем случае, рекомендуется поступать следующим образом. Весь проект следует строить согласно методологии, разработанной компанией Protel, с применением оформленных по ГОСТ символов компонентов, русскоязычных имен цепей и листов. В этом случае пользователь может максимально использовать все функциональные возможности системы Protel 99 SE для сквозного проектирования, включая корректную генерацию списков соединений, проверку правил электрических соединений (ERC) и т.д. В этом случае все схемы проекта будут оформлены "почти по ГОСТ".

Подготовку схем для печати так, чтобы они полностью соответствовали требованиям ГОСТ, следует производить на окончательном этапе проектирования, причем возможно даже на специально созданных для этого копиях. Подготовку к печати можно делать вручную и автоматически с помощью предназначенных для этой цели макросов. На этом этапе можно заменить все встроенные изображения портов на стандартные изображения разъемов с копированием и комбинированием различной текстовой информации (адреса), удалить лишние метки цепей, выключить позиционные обозначения многосекционных микросхем, где в качестве разделителя номера секции используется двоеточие, и вместо них вставить текстовые поля с корректным обозначением (разделитель — точка), а также много другое.

Такой подход позволит максимально эффективно работать независимо от того, требуется вывод оформленных по ГОСТ схем на бумагу для архива или нет.

Глава 4

Моделирование смешанных сигналов

Моделирование смешанных сигналов — основные возможности

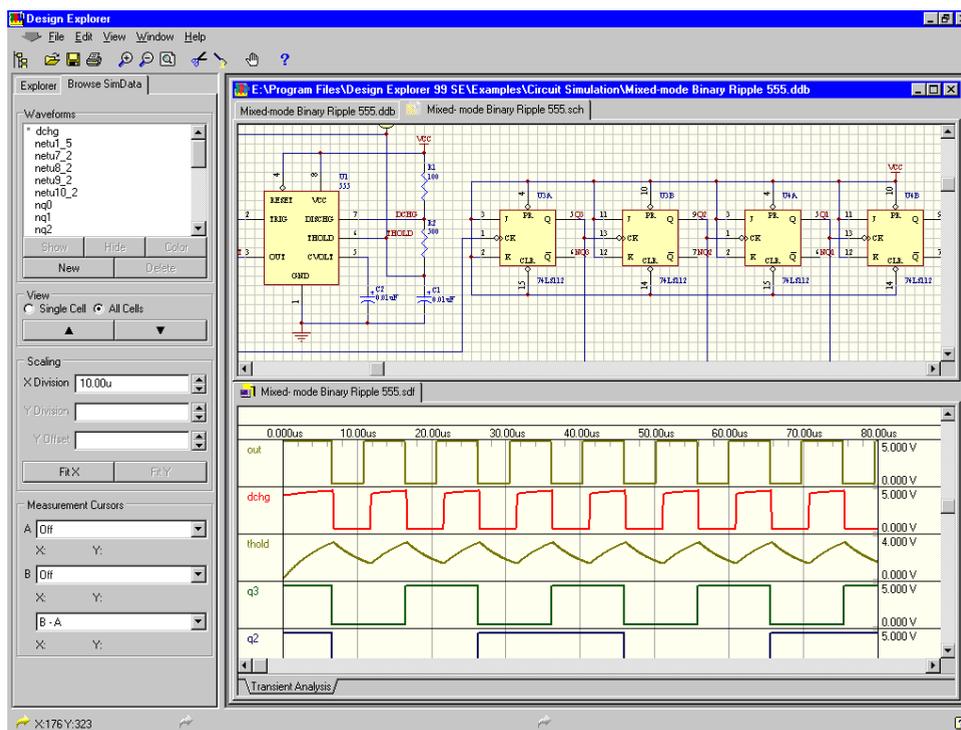


Рис. 4.1. Моделирование в системе Protel 99 SE.

В данной главе содержатся сведения по моделированию поведения аналоговых и цифровых схем. Программа моделирования полностью интегрирована в редактор принципиальных схем, то есть запуск процесса моделирования может быть произве-

ден непосредственно из окна схемного редактора сразу по окончании разработки схемы.

В этой главе пользователь сможет найти всю необходимую информацию по установке параметров и запуску процесса моделирования, а также изучить основные возможности системы для эффективного проведения различных типов анализа, отображения результатов и работы с ними. Данная глава может быть полезна пользователям системы P-CAD 2001, моделирование в которой выполняется в среде Design Explorer.

Создание проекта

Прежде, чем приступить к моделированию, необходимо ввести схему с помощью редактора принципиальных схем. Программе моделирования требуется специальная информация (например, тип компонента, используемая модель и т.д.) о каждом элементе схемы. Эта информация хранится в библиотеках символов элементов, предназначенных именно для моделирования.

Полный набор библиотек находится в библиотечной базе данных в каталоге `\Program Files\Design Explorer 99 SE\Library\Sch\Sim.ddb`. Все символы, находящиеся в этих библиотеках, связаны с соответствующими моделями. Существует возможность, по окончании разработки схемы, произвести необходимые установки и запустить процесс моделирования непосредственно из окна редактора схем. В любой момент можно остановить процесс, внести изменения в установки и повторить запуск.

Передовая технология моделирования

Система Protel 99 SE позволяет производить точное, реалистичное моделирование аналоговых, цифровых и смешанных схем. Результаты компьютерного анализа, как правило, идентичны результатам, получаемым при макетировании, а смоделированное поведение устройств в точности повторяет функционирование реального изделия. Например, цифровые интегральные схемы имеют задержку распространения, времена установки и удержания, учитываются нагрузки на всех выводах устройств, то есть в расчете учитываются почти все реальные параметры.

Для разностороннего тестирования и анализа схемы пользователю предоставляется широкий выбор вариантов моделирования.

Совместимость с пакетом SPICE

Программа моделирования использует расширенную версию пакета Berkeley SPICE3f5/Xspice, которая позволяет точно моделировать любую комбинацию из аналоговых и цифровых устройств, что стало возможным благодаря использованию точных управляемых событиями поведенческих моделей цифровых устройств, включая TTL и КМОП логику.

Использование SimCode для цифрового моделирования

Программа моделирования выполняет истинное моделирование смешанных сигналов, это означает, что могут анализироваться как цифровые, так и аналоговые устройства. Однако, учитывая сложность цифровых устройств, практически не возможно моделировать их, используя стандартные (не управляемые событиями) команды SPICE. По этой причине в программу моделирования включен специальный язык

описания, который позволяет, при использовании расширенной версии XSPICE (поддерживающей управление событиями), моделировать цифровые устройства. Цифровые устройства, включенные в библиотеки моделей, смоделированы с помощью патентованного языка Digital SimCode™, специально разработанным для использования с программой моделирования из пакета Protel 99 SE.

Поддержка моделей производителей

Программа моделирования поддерживает модели от таких производителей, как Motorola, Texas Instruments и др., которые создают модели для обеспечения максимальной совместимости с аналоговым моделированием. Система дает возможность использовать эти модели непосредственно, без дополнительной адаптации.

Полный набор библиотек моделей

Программа моделирования включает в себя полный набор библиотек, находящийся в базе данных в папке `\Program Files\Design Explorer 99 SE\Library\Sch\Sim.ddb`. Каждый элемент этих библиотек готов к использованию. При размещении элемента на листе принципиальной схемы происходит автоматическое установление связи с соответствующей моделью. Специальное подразделение компании Altium занимается постоянным созданием новых и обновлением уже имеющихся библиотек для всех продуктов Protel. Самую последнюю версию библиотек можно найти по адресу www.protel.com.

Ограничения моделирования

Программа моделирования не имеет ограничений при моделировании аналоговых устройств на схемотехническом уровне и при моделировании цифровых устройств на уровне вентилях. Схема может состоять как из одного, так и из нескольких листов, иметь сложную иерархию, причем размер схемы ограничивается только размером доступной вам оперативной памяти.

Используемые типы анализа

Программа моделирования поддерживает большое количество типов анализа, включая частотный анализ в режиме малого сигнала, анализ переходных процессов, анализ шумов, а также анализ передаточных функций по постоянному току. Кроме вышеперечисленных базовых методов анализа, также имеется возможность проведения статистического анализа методом Monte-Carlo, анализа с изменением значений параметров и температуры, и наконец анализа Фурье.

Математические действия с рассчитанными сигналами

Система имеет возможность математической обработки рассчитанных сигналов, то есть их сложения, вычитания, применения к ним различных математических функций. Полученные таким образом новые зависимости могут быть отображены в специальном окне, как и любые другие сигналы.

Зависимые источники

Программа моделирования содержит модели источников сигналов имеющих линейные и нелинейные зависимости. как правило, они используются для построения эквивалентных схем различных устройств, рассматриваемых как “черный ящик”.

Линейные зависимые источники

Устройства **E**, **F**, **G** и **H** имеют встроенные выражения для задания простой линейной зависимости. Такие устройства предлагают легкий способ моделирования простых линейных эффектов.

Нелинейные зависимые источники

Эти устройства позволяют задать выражение для определения величины напряжения или тока с использованием различных нелинейных функций (log, ln, exp, sin и т. д.), наиболее часто встречающихся в электрических схемах.

Введение в моделирование

После создания схемы из библиотечных элементов, прежде чем запустить процесс моделирования, необходимо произвести три простых действия:

1. Добавить соответствующий источник сигнала.
2. Определить контрольные точки.
3. Установить параметры моделирования.

Теперь пользователь готов к моделированию схемы, однако, рекомендуется ознакомиться с большим числом прилагаемых примеров моделирования, сохраненных в виде обычных баз данных проекта.

Примеры схем

Программа моделирования включает большое количество примеров схем, которые демонстрируют различные виды и способы анализа, а также показывают использование различных типов устройств.

Кроме этого, примеры дают возможность рассмотреть различные варианты установки параметров моделирования. Более подробная информация о конфигурировании различных типов анализа, необходимая при изучении приведенных примеров, приведена в разделе *Подготовка и проведение моделирования*.

Создание схемы

Пакет Protel 99 SE содержит приблизительно 5800 аналоговых и цифровых элементов схем, связанных с соответствующими моделями и полностью готовых к моделированию. Все эти компоненты соответствуют различным конструктивным исполнениям и топологическим посадочным местам (где это возможно).

Размещение компонентов

Библиотеки элементов схем, предназначенных для моделирования, расположены в папке `\Program Files\Design Explorer 99 SE\Library\Sch\Sim`. За информацией по поиску компонентов в библиотеках обратитесь к подразделу *Библиотеки символов компонентов* раздела *Компоненты и модели*.

Установка обозначений компонентов

В системе Protel 99 SE не существует никаких ограничений на использование префиксов обозначений компонентов. Например, интегральные микросхемы можно обозначать **U1**, **U2** или **IC1**, **IC2** и т. д.

Множители, используемые при задании параметров компонентов

Система Protel позволяет задавать параметры элементов с использованием буквенных множителей, так например "n" означает "нано", "k" — "кило", и т. д. Допустимые множители представлены в таблице, все остальные буквы игнорируются. Любые буквы, следующие после множителя, также игнорируются. Необходимо, чтобы множитель находился сразу после соответствующей цифры, наличие пробелов между множителем и цифрой недопустимо. Ввод множителя может производиться в верхнем или нижнем регистре. Ниже представлены 3 примера:

- **10**, **10V**, **10Volts**, и **10Hz** представляют одно и то же число — **10**. Во всех случаях буквы игнорируются, так как ни одна из них не является допустимым множителем.
- **M**, **m**, **MA**, **MSec**, и **MMhos** представляют один и тот же множитель **10⁻³**. В этом случае все буквы после первой "m" игнорируются.
- **1000**, **1000.0**, **1000Hz**, **1e3**, **1.0e3**, **1KHz**, и **1K** — все представляют число **1000**.

Таблица 4.1

Буквенное обозначение	Множитель
T	10^{12}
G	10^9
Meg	10^6
K	10^3
mil	25.4^{-6}
m	10^{-3}
u (or μ)	10^{-6}
n	10^{-9}
p	10^{-12}
f	10^{-15}

Ввод источников сигналов

Перед запуском процесса моделирования необходимо ввести необходимые источники сигналов и генераторы тактовых импульсов. Это можно сделать с помощью команды меню **Simulate » Sources**. При размещении, выбрав наиболее подходящий источник из меню, можно тут же настроить его параметры. Возможно также размещать их непосредственно из библиотеки элементов моделирования, которая находится в папке **Program Files\Design Explorer 99 SE\Library\Sch\Sim.ddb**.

В программу моделирования включены как независимые источники постоянного, синусоидального, импульсного, экспоненциального, частично-линейного и частотно-модулированного сигналов, так и зависимые линейные или нелинейные источники.

Для установки параметров необходимо произвести двойной щелчок левой кнопкой мыши на размещенном источнике сигналов. За более подробной информацией по конфигурированию источников сигналов обратитесь к разделу *Источники напряжения и тока*.

Установка точек контроля сигналов

Определение точек контроля сигналов, происходит в диалоговом окне **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup**. В верхней части вкладки **General** помещены два списка: доступные переменные (**Available Variables**) и активные переменные (**Active Variables**). Для каждой переменной из списка **Active Variables** форма сигналов будет отображаться автоматически. Не имеет значения, что сейчас переменная не включена в список, так как можно добавлять и удалять переменные непосредственно из окна отображения сигналов, после его появления.

За более подробной информацией по отображению сигналов обратитесь к подразделу *Сбор, отображение и сохранение результатов моделирования* раздела *Подготовка и проведение моделирования*.

Установки параметров анализа

Все установки параметров анализа производятся в диалоговом окне **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup**.

Диалоговое окно открывается на вкладке **General**, где производится выбор вида анализа. Конфигурирование каждого конкретного вида анализа производится на других вкладках данного диалогового окна. За более подробной информацией по конфигурированию каждого из типов моделирования обращайтесь к разделу *Подготовка и проведение моделирования*.

Кнопка **Advanced** в диалоговом окне **Analyses Setup** позволяет получить доступ к SPICE переменным. Необходимость изменения этих дополнительных параметров может появиться только при возникновении трудностей в моделировании схемы или еще по какой-либо особой причине. За более подробной информацией относительно этих переменных обратитесь к разделу *Переменные и параметры модуля SPICE*. При возникновении трудностей с моделированием схем обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Запуск процесса моделирования

После того как все параметры сконфигурированы, система моделирования готова к запуску. Запустить моделирование можно одним из следующих способов: нажатием кнопки **Run Analyses** в диалоговом окне **Analyses Setup**, выбором команды меню **Simulate » Run** или нажатием кнопки **Run** в панели инструментов **Simulation Tools**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается, и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает

возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Просмотр результатов моделирования

По завершению процесса моделирования результаты автоматически появляются в специальном окне просмотра результатов. Для просмотра результатов расчета каждого конкретного типа анализа необходимо щелкнуть на соответствующей вкладке. С помощью органов управления масштабированием, расположенных на панели управления, можно настроить вид отображения результатов. Для включения этой панели необходимо воспользоваться командой меню **View » Panel**.

За более подробной информацией по использованию органов масштабирования и измерения обратитесь к разделу *Окно просмотра результатов расчета*.

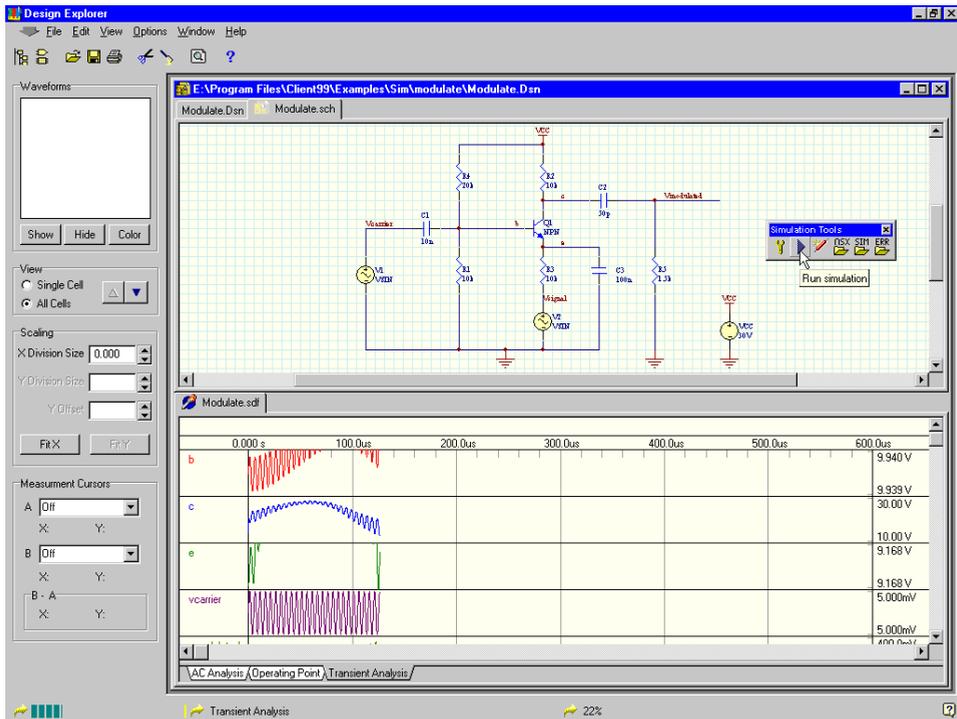


Рис. 4.2. В строке состояния отображается ход выполнения моделирования, а на графиках — рассчитанные сигналы.

Подготовка и проведение моделирования

Для вызова диалогового окна **Analyses Setup**, в котором можно установить все доступные виды анализа и их параметры, необходимо воспользоваться командой **Simulate » Analyses Setup**. Диалоговое окно **Analyses Setup** содержит главную вкладку **General**, где производится выбор типов анализа, и задаются опции отображения результатов расчета, а также другие вкладки для конфигурирования каждого конкретного вида анализа.

Как только создание схемы завершено, процесс моделирования готов к запуску. Запустить моделирование можно одним из следующих способов: нажатием кнопки **Run Analyses** в диалоговом окне **Analyses Setup**, выбором команды меню **Simulate » Run**, или нажатием кнопки **Run** в панели инструментов **Simulation Tools**. Для уточнения деталей по конфигурированию программы моделирования для выполнения отдельных видов анализа обратитесь к соответствующим разделам данной главы.

Всякий раз, при запуске моделирования, строка состояния отображает ход процесса. Первое, что происходит по команде меню **Simulate » Run**, это создание списка соединений схемы. После окончания создания списка соединений, он передается программе моделирования, которая отображает результаты моделирования по мере их получения (рис. 4.2).

Как только процесс моделирования завершается, результаты появляются в окне отображения результатов моделирования. Это окно содержит вкладки для каждого из произведенных видов анализа. За более подробной информацией по работе с результатами моделирования обратитесь к разделу *Окно просмотра результатов расчета*.

Сбор, отображение и сохранение результатов моделирования

В нижней части вкладки **General** диалогового окна **Setup Analyses** содержит два списка, позволяющих определить, какие из результатов расчета будут сохраняться в файле, а какие сигналы отображаться в специальном окне (рис. 4.3).

Сбор результатов

Список **Collect Data For** однозначно задает вид информации, которая должна быть обработана и сохранена в специальном файле. Нужная опция выбирается из пяти доступных в выпадающем списке. Выбор одной из первых четырех опций позволяет задать тип информации, которая будет рассчитана (напряжение, ток, мощность, импеданс и т. д.) и сохранена в файле результатов.

Для каждой из первых четырех опций сохраняются результаты для всех доступных сигналов в схеме. Пятая опция предписывает программе моделирования сохранять данные только для сигналов, указанных в списке активных сигналов **Active Signals**. Использование этой опции позволяет минимизировать размер файла результатов.

Список **Active Signals** используется также для ограничения числа узлов, обрабатываемых во время циклических видов анализа, например, при статистическом анализе методом Monte-Carlo, при автоматическом изменении параметров элементов или при автоматическом изменении температуры. В случае, если опция **Active Signals** не выбрана, то при этих типах моделирования результаты собираются для всех узлов схемы.

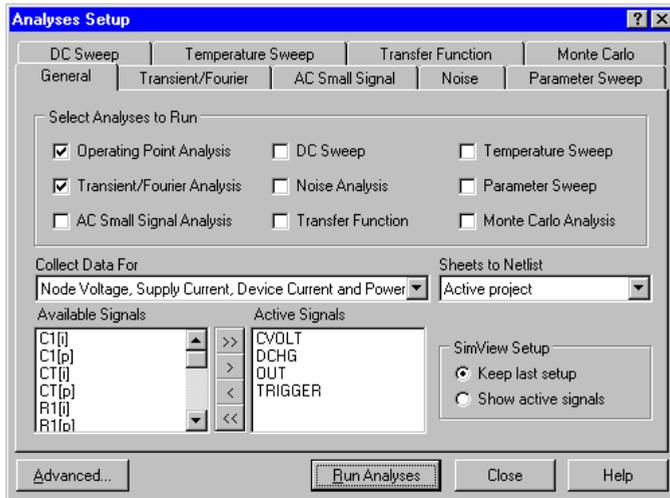


Рис. 4.3. Вкладка General окна настройки типов анализа.

Списки Signals и Active Signals

В списке **Available Signals** показываются все переменные в схеме соответствующие строке в списке **Collect Data For**. Программа моделирования будет сохранять результаты для всех переменных показанных в списке **Available Signals**.

Все переменные, указанные в списке **Active Signals**, будут автоматически отображаться в окне результатов моделирования.

Опции SimView Setup

При запуске процесса моделирования результаты отображаются в окне просмотра результатов, вид которого может быть настроен в соответствии с потребностями пользователя. В частности, существует возможность изменения масштаба изображения, добавление и удаление контрольных точек и т. д. Вся информация о настройках окна просмотра результатов расчета сохраняется в файле с расширением **.SDF**.

Информация о настройках, по умолчанию, сохраняется таким образом, что повторное открытие файла результатов моделирования позволяет видеть данные в том же виде, в каком они были до закрытия. При повторном запуске процесса моделирования, для отображения вновь сконфигурированных результатов и назначенных контрольных

точек, необходимо выключить опцию **Keep last setup** и включить опцию **Show active signals**.

Анализ переходных процессов (Transient Analysis)

Сигнал, получаемый в результате анализа переходных процессов, подобен тому, что мы обычно видим на экране осциллографа. Этот сигнал является результатом вычисления переходных переменных (напряжения или тока) как функции времени на определяемом пользователем временном интервале. Вне зависимости от использования опции, определяющей начальные условия (**Use Initial Conditions**), для определения постоянной составляющей в схеме перед моделированием переходных процессов, автоматически производится расчет рабочих точек (**Operation Point Analysis**). Подробности по вопросу задания и использования начальных условий приведены в разделе *Установка начальных условий*.

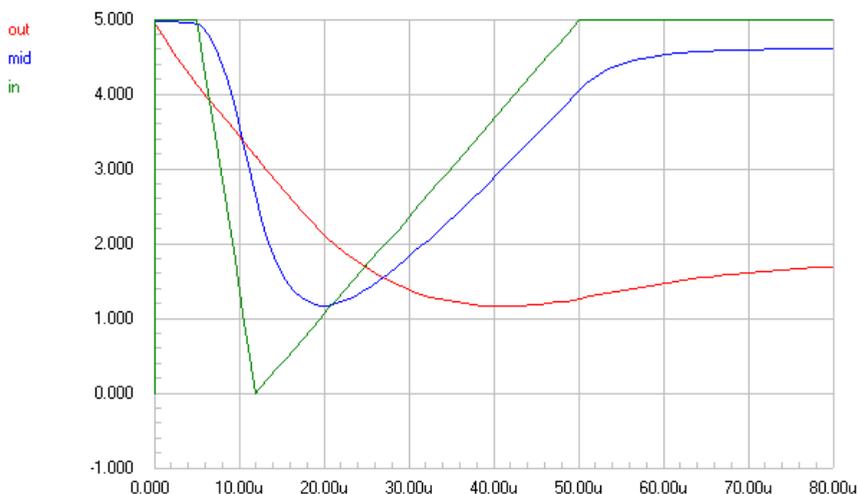


Рис. 4.4. При анализе переходных процессов рассчитываются токи и напряжения, как функция времени

Установка параметров для анализа переходных процессов (Transient Analysis)

Параметры моделирования переходных процессов устанавливаются на вкладке **Transient/Fourier** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate** » **Analyses Setup** (рис. 4.5).

Анализ переходных процессов всегда начинается в нулевой момент времени. Если значение параметра **Start Time** не совпадает с нулевым моментом времени, программа

производит анализ схемы в этом промежутке времени, но полученные результаты не отображаются и не сохраняются. В промежутке времени между моментом запуска моделирования (**Start Time**) и моментом остановки процесса (**Stop Time**) анализ схемы продолжается, при этом результаты расчета сохраняются и отображаются в окне просмотра.

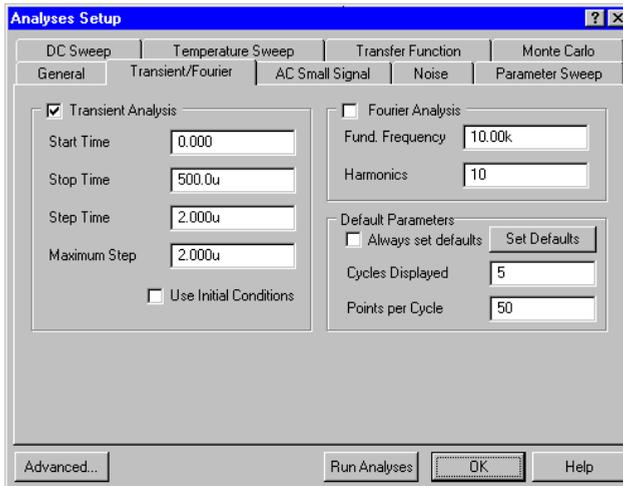


Рис. 4.5. Настройка анализа переходных процессов.

Временной шаг (**Step Time**) соответствует номинальному, используемому при моделировании, однако, реальный шаг изменяется автоматически программой моделирования для достижения сходимости. Значение **Maximum Step** ограничивает диапазон изменений временного шага, который используется программой моделирования при вычислении переходных данных. По умолчанию, это значение выбирается меньшим из **Step Time** или $(\text{Stop Time} - \text{Start Time})/50$. Как правило, **Step Time** и **Maximum Step** имеют одни и те же значения.

Автоматическое вычисление параметров анализа переходных процессов

В случае, если пользователь не уверен в правильности вводимых значений можно воспользоваться кнопкой **Set Defaults** для автоматического вычисления параметров анализа переходных процессов. В этом случае значение **Start Time** устанавливается равным нулю, а параметры **Stop Time**, **Step Time** и **Maximum Step** вычисляются с использованием установок по умолчанию. Параметры отображения тактов (**Cycles Displayed**), а также количество точек на такт (**Points per Cycle**) основываются на самой низкой из частот источников сигналов присутствующих в схеме. Например, если самая низкая частота источника сигналов равна 10кГц и значения параметров по умолчанию равны **Cycles Displayed = 5**, а **Points per Cycle = 50**, то:

$$\begin{aligned}\text{Stop Time} &= 1/10\text{KHz} * 5 \\ &= 100\mu\text{S} * 5 \\ &= 500\mu\text{S}\end{aligned}$$

$$\begin{aligned}\text{Step Time} &= (1/10000) / 50 \\ &= 2\mu\text{S}\end{aligned}$$

$$\begin{aligned}\text{Max Step} &= \text{Step Time} \\ &= 2\mu\text{S}\end{aligned}$$

Если при анализе переходных процессов всегда установлена опция для значений по умолчанию, программа моделирования функционирует так, как если бы кнопка **Set Defaults** нажималась перед каждым стартом процесса моделирования.

Необходимость использования опции задания начальных условий

При включении опции **Initial Conditions Option** моделирование переходных процессов начинается с заданных начальных условий, минуя этап расчета рабочих точек. Эта опция позволяет начинать моделирование переходных процессов со значений, отличающихся от статических рабочих точек.

Для использования этой опции необходимо или определить начальные условия для каждого компонента схемы, или разместить на схеме специальные устройства **.IC**, определяющие начальные условия. Если у компонента начальные условия не определены, они берутся равными нулю. Для редактирования соответствующего поля **Part Field** необходимо выполнить двойной щелчок левой кнопкой мыши на соответствующем элементе. Более подробная информация по заданию начальных условий для каждого конкретного типа элементов приведена в разделе *Компоненты и модели*. Альтернативной возможностью задания начальных условий является размещение специальных устройств **.IC** из библиотеки **Simulation Symbols.lib**.

Обычно, лучшим способом задания начальных условий для анализа переходных процессов является использование устройств **.IC**. Более подробная информация по использованию устройств **.IC** представлена в разделе *Установка начальных условий*. Следует отметить, значения начальных условий, заданных для компонентов имеют более высокий приоритет и переопределяют устройства **.IC**, присоединенные к соответствующим цепям.

Запуск анализа переходных процессов

Для запуска моделирования переходных процессов необходимо:

1. Установить параметры для режима **Transient Analysis**, как описано выше.
2. Включить опцию **Transient/Fourier Analysis** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
3. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает

возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета.

В этом окне можно просмотреть результаты и уточнить отдельные значения для таких параметров, как напряжение, ток, рассеиваемая мощность и т. д. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

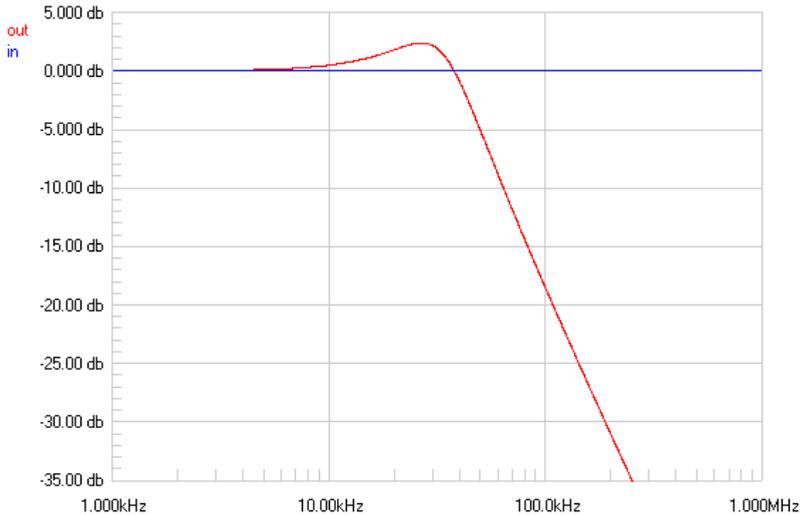


Рис. 4.6. Частотный анализ схемы.

Частотный анализ схемы в режиме малого сигнала (AC Sweep)

Данные, получаемые в результате анализа схемы в режиме малого сигнала представляют собой частотные характеристики схемы, рассчитанные с использованием мало-сигнальных моделей элементов. Процесс моделирования начинается с расчета рабочих точек для определения режима по постоянному току, затем производится замена источников сигналов генераторами синусоидального сигнала с фиксированной амплитудой, и наконец производится анализ в заданном частотном диапазоне. Искомые результаты обычно представляются в виде передаточной функции, например, коэффициент усиления по напряжению.

Установка параметров для частотного анализа в режиме малого сигнала

Параметры для частотного анализа в режиме малого сигнала устанавливаются на вкладке **AC Small Signal** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate** » **Analyses Setup** (рис. 4.7).

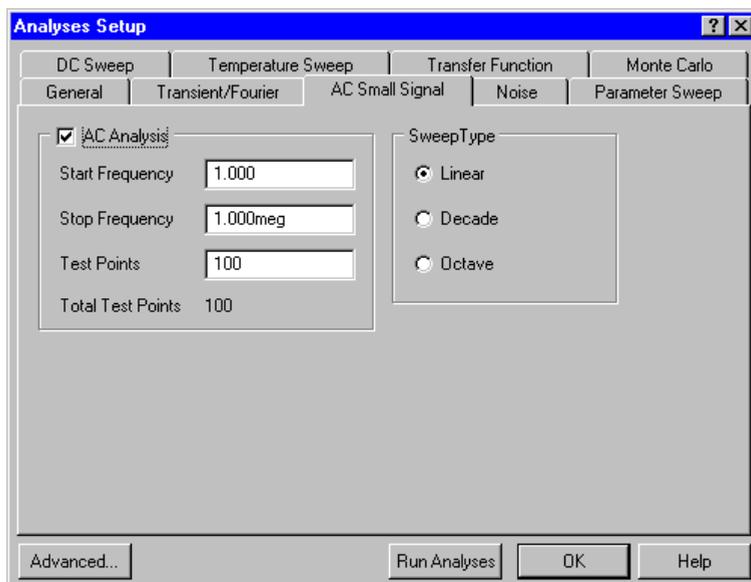


Рис. 4.7. При частотном анализе схемы в режиме малого сигнала рассчитываются ее амплитудно- и фазочастотные характеристики.

Как минимум один из источников, присутствующих в схеме, должен содержать значение в поле **AC Part Field**. Во время моделирования этот источник заменяется генератором синусоидального сигнала, при этом производится перестройка его частоты в диапазоне от **Start Frequency** до **Stop Frequency** с шагом, определяемым значениями **Test Point** и **Sweep Type**.

Необходимо установить тип изменения сигнала (**Sweep Type**) для определения тестовых точек. Опции изменения приведены ниже:

- **Linear** — общее число частотных точек;
- **Decade** — число частотных точек на декаду;
- **Octave** — число частотных точек на октаву.

Амплитуда и фаза изменяемого синусоидального сигнала задаются в соответствующих полях **Part Field** источника. Для задания этих значений необходимо произвести двойной щелчок левой кнопкой мыши на нужном источнике. Значение амплитуды вводится в поле **AC Part Field** (в вольтах), а значение фазы вводится в поле **AC Phase Part Field** (в градусах). Указание единиц измерения не требуется.

Запуск частотного анализа в режиме малого сигнала

Для запуска частотного анализа в режиме малого сигнала необходимо:

1. Установить параметры для режима **AC Small Signal**, как описано выше.
2. Убедиться, что по крайней мере, один источник сигнала содержит значение в поле **AC Part Field**.
3. Включить опцию **AC Small Signal Analysis** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
4. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета.

В этом окне можно просмотреть результаты и уточнить отдельные значения для таких параметров, как напряжение, ток, рассеиваемая мощность и т. д. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Анализ схем при изменяющемся постоянном напряжении (DC Sweep)

В результате анализа схем при изменяющемся постоянном напряжении источника (**DC Sweep**) получаются данные, аналогичные получаемым с помощью характеристических графиков. В процессе анализа производится последовательный расчет серии рабочих точек, изменяется значение напряжения выбранного источника постоянного напряжения с заранее заданным шагом и строится соответствующая кривая (рис. 4.8). Кроме того, имеется возможность определить дополнительный (второй) изменяемый источник постоянного напряжения.

Установка параметров DC Sweep анализа

Установка параметров для **DC Sweep** анализа производится на вкладке **DC Sweep** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup** (рис. 4.9).

В списке **Source Name** задается имя независимого источника постоянного напряжения, параметры которого должны изменяться с заданным шагом. Значения **Start Value**, **Stop Value**, **Step Value** определяют диапазон и шаг изменения напряжения. Если определен второй источник постоянного напряжения, параметры основного источника изменяются с заданным шагом во всем диапазоне для каждого из значений параметров этого источника питания.

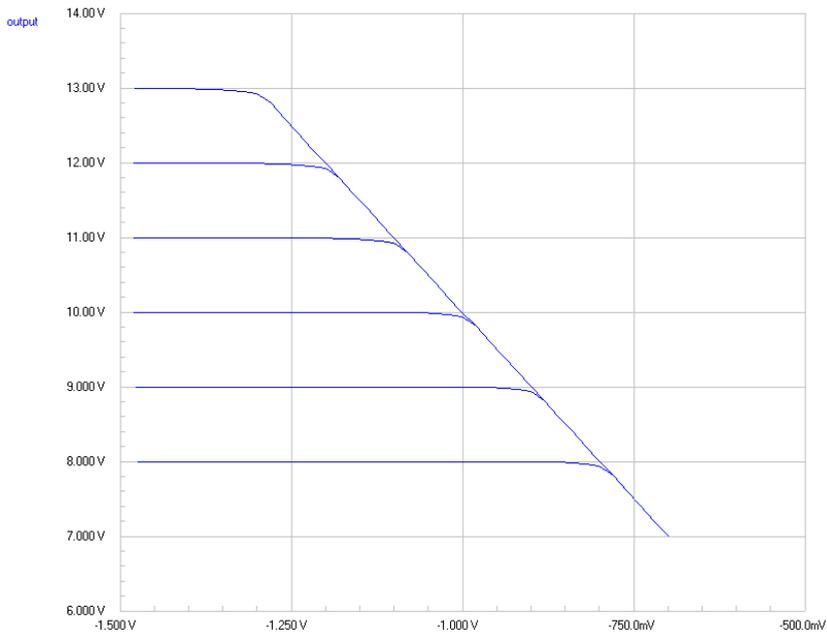


Рис. 4.8. При анализе схем в режиме DC Sweep строится семейство характеристик, например, при ступенчато изменяемом напряжении смещения.

Запуск DC Sweep анализа

1. Установите параметры **DC Sweep** анализа, как описано выше.
2. Включите опцию **DC Sweep**, на вкладке **General** диалогового окна **Analyses Setup**, вызываемого с помощью команды меню **Simulate » Analyses Setup**.
3. После включения этой опции, для запуска процесса моделирования можно воспользоваться клавишей **Run Analyses** в нижней части диалогового окна или вызвать пункт меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается, и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

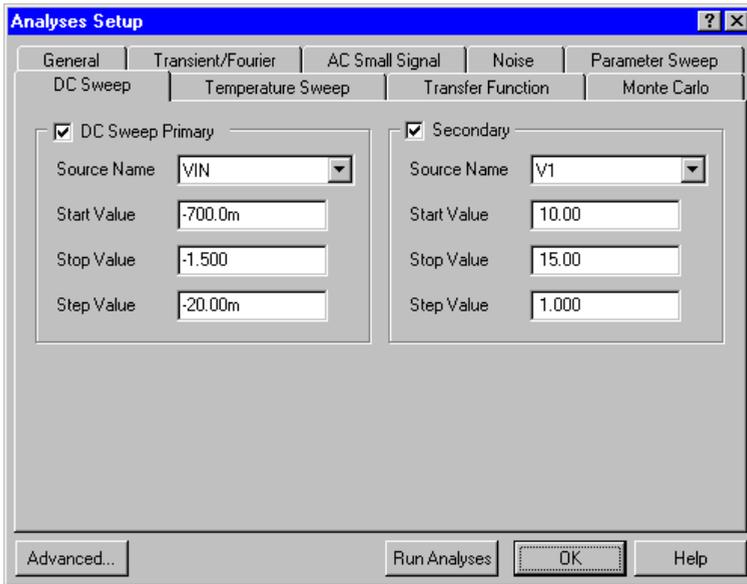


Рис. 4.9. Настройка анализа DC Sweep.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они появляются по мере расчета.

В окне отображения результатов моделирования есть возможность просматривать рассчитанные кривые и уточнять отдельные их значения. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Расчет рабочей точки по постоянному току (DC Operating Point)

Перед выполнением расчета в режимах **Transient** или **AC Small Signal** необходимо произвести расчет рабочих точек по постоянному току (**Operation Point Analysis**). Во время этого вида расчета определяется смещение по постоянному току (**DC bias**) для схемы в целом, при этом эквивалентом индуктивности является перемычка, а эквивалентом емкости — разрыв цепи. Для всех нелинейных элементов схемы определяются линеаризованные малосигнальные модели, которые будут использоваться при проведении расчета частотных характеристик устройства в режиме малого сигнала

(**AC Small Signal Analysis**). Все источники переменного напряжения при этом не учитываются.

Запуск Operating Point анализа

Расчет рабочих точек производится автоматически всякий раз, когда запускается моделирование **Transient** или **AC Small Signal**. Результаты расчета используются программой моделирования для внутренних целей. Для проверки результатов вычисления рабочих точек необходимо:

1. Включить опцию **Operating Point Analysis** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
2. Для вычисления значений тока, мощности и напряжения необходимо выбрать соответствующую опцию в выпадающем списке **Collect Data For** на вкладке **General** диалогового окна **Analyses Setup (Node Voltage, Supply Currents, Device Currents и Powers)**.
3. После установки этой опции, запуск процесса моделирования возможен нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается, и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информации обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Результаты расчета показываются на вкладке **Operating Point** окна отображения. Существует возможность добавления сигналов на отображение, для чего на вкладке просмотра необходимо выбрать нужный сигнал и нажать кнопку **Show**.

Статистический анализ методом Monte Carlo

При статистическом анализе методом Monte Carlo происходит многократный запуск процесса моделирования, при этом параметры элементов принимают случайные значения в заданных пределах вблизи указанного (рис. 4.10). Использование этой функции возможно только при условии проведения одного (или более) из стандартных видов анализа (**AC**, **DC** или **Transient**). Результаты расчета методом Monte Carlo могут быть получены только для узлов, которые помещены в список **Active Variable** диалогового окна **Setup Analyses**.

Параметры подсхем во время анализа методом Monte Carlo изменены быть не могут. Изменяться могут только параметры базовых компонентов и моделей.

Установка параметров для анализа методом Monte Carlo

Установка параметров для статистического анализа методом Monte Carlo производится на вкладке **Monte Carlo** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup** (рис. 4.11).

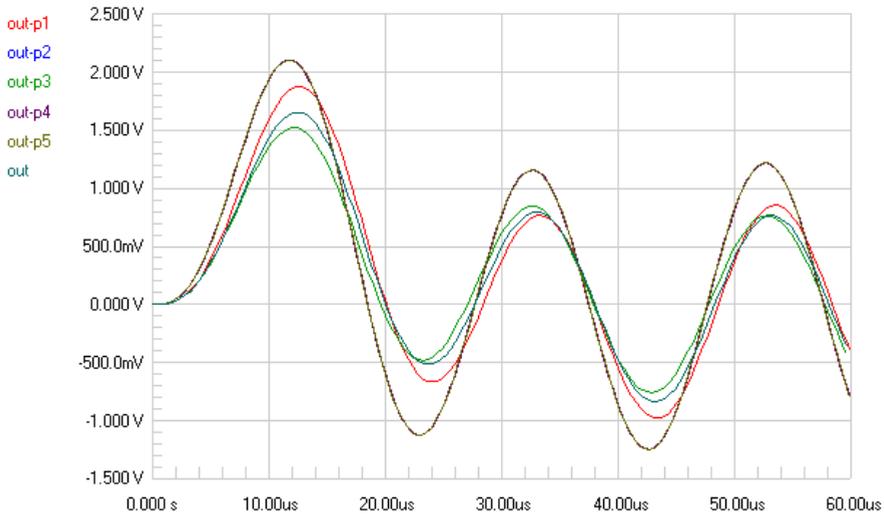


Рис. 4.10. При статистическом анализе методом Monte Carlo строится семейство характеристик, полученных при случайном изменении параметров схемы в заданных пределах.

Здесь устанавливаются следующие опции:

Simulation Runs (число запусков расчета)

Эта переменная задает требуемое число запусков процесса моделирования. Например, если задано число 10, то будет произведено 10 запусков процесса моделирования. При этом для каждого запуска будут использованы различные значения параметров элемента из заданного диапазона.

Simulation Seed (начальное значение генератора случайных чисел)

При анализе методом Monte Carlo для генерации псевдослучайных значений параметров необходимо задать некое начальное число. По умолчанию это число задается равным -1. Для реализации различных псевдослучайных последовательностей при нескольких запусках этого вида анализа начальное число необходимо изменять.

Default Distribution (вид распределения)

Для генератора псевдослучайных чисел, используемого в процессе моделирования по методу Monte Carlo, имеется возможность выбора одного из трех видов распределения:

Uniform Distribution (равномерное распределение)

При этом типе распределения значения параметров равномерно распределяются в заданном диапазоне. Например, для резистора в 1К с допуском $\pm 10\%$ генерация значения в диапазоне от 900 Ом до 1100 Ом равновероятна.

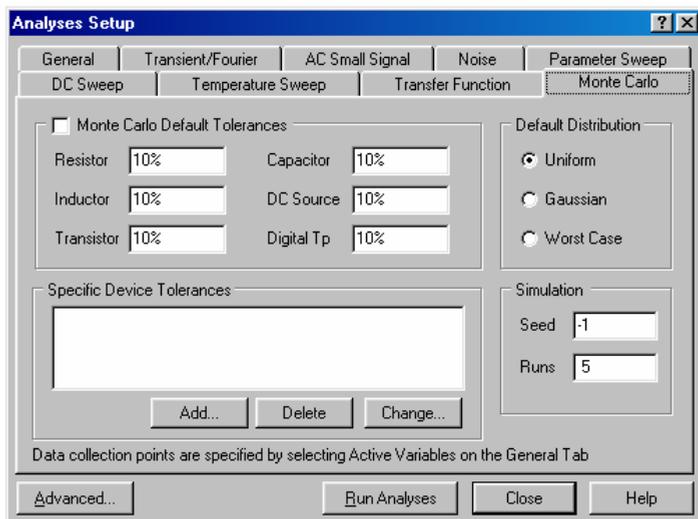


Рис. 4.11. Настройка анализа методом Monte Carlo.

Gaussian Distribution (Гауссовское распределение)

Значения распределяются согласно Гауссовской кривой, с центром соответствующим номинальному значению и заданным допуском в ± 3 стандартного отклонения. Для резистора в 1К с допуском $\pm 10\%$ центр распределения будет равен 1000 Ом, $+3$ стандартных отклонения будет соответствовать 1100 Ом, а -3 стандартных отклонения — 900 Ом. Этот тип распределения дает наибольшую вероятность генерации значения близкого к номинальному значению.

Worst Case (худший случай)

Этот тип распределения похож на равномерное распределение, только используются крайние точки диапазона (наихудший вариант). Для резистора в 1К с допуском $\pm 10\%$ будут произвольно генерироваться значения из двух наихудших вариантов — 900 Ом и 1100 Ом.

Default Tolerances (допуски по умолчанию)

Имеется возможность задания допусков для шести основных категорий компонентов: резисторов, конденсаторов, катушек индуктивности, источников постоянного напряжения, транзисторов (коэффициент передачи по току), задержек распространения (в цифровых устройствах).

Допуски могут задаваться как абсолютными, так и относительными (в процентах) значениями. Например, допуск резистора можно определить как 10 или как 10%. Если допуск резистора сопротивлением 1К определен как 10, это означает, что его сопротивление может изменяться в диапазоне от 990 до 1010 Ом. При определении допуска 10% сопротивление принимает значения в диапазоне от 900 до 1100 Ом.

Значения параметров каждого элемента изменяются независимо от остальных. Например, если в схеме есть два резистора сопротивлением 10К, а допуск установлен равным 10%, то во время первого прохождения процесса моделирования один из резисторов может получить значение 953 Ом, а другой 1022 Ом. Программа моделирования использует независимые генераторы псевдослучайных чисел для получения значений параметров каждого отдельного элемента.

Specific Device Tolerance (специфические допуски)

Существует возможность переопределения значений допусков, используемых по умолчанию (**Default Tolerances**). Для задания специфических допусков элементов (**Specific Device Tolerance**) необходимо щелкнуть правой кнопкой мыши в поле **Specific Device Tolerance** и в появившемся контекстном меню выбрать опцию **Add** (или нажать горячую клавишу **INSERT**). Появится диалоговое окно Monte Carlo с вкладкой **Device and Lot Tolerances**, в котором задаются следующие параметры:

Designator

Здесь из выпадающего списка выбирается нужный элемент схемы.

Parameter

Сюда вводится значение параметра элемента, если оно еще не определено. Здесь можно ввести, например, задержку распространения для цифрового элемента, коэффициент передачи по току для транзисторов и сопротивление для резисторов.

Device Tolerance

Допуски на значение параметра данного элемента.

Device Tracking Number (трековый номер элемента)

Присвоение одинакового трекового номера нескольким разным элементам производится в случае, когда необходимо коррелированное изменение их параметров. Если двум элементам назначить одинаковые трековые номера (**Device Tracking Number**) и типы распределения (**Device Distribution**), то в процессе моделирования, для обоих элементов будут использованы одни и те же псевдослучайные значения.

Device Distribution

Позволяет выбрать вид распределения для данного элемента.

Lot Tolerance, Tracking and Distribution

Эти установки аналогичны таковым из поля **Device**. Они обеспечивают дополнительный способ определения допусков элементов и их корреляции. Система поддерживает как допуски на параметры отдельных элементов (**Device Tolerance**), так и допуски на группы элементов (**Lot Tolerance**), но необходимым для процесса моделирования является только один из них. Программа моделирования рассчитывает допуски элементов и групп независимо (используя различные псевдослучайные числа), а затем объединяет их вместе.

Объединение допусков элементов и групп бывает полезным в случае, когда их значения оказываются не совсем коррелированными, но в то же время не являются полностью независимыми. Например, есть две резисторные сборки. В этом случае допуск группы (который определяется от сборки к сборке) может быть большим, в то же время допуск элементов (по отношению к резисторам, находящимся в одной сборке)

будет малым. Таким образом, допуск элементов не может быть проигнорирован, так как это может исказить общую характеристику работы схемы.

Рассмотрим следующий пример:

Примем R1 и R2 равными 1К с допуском элементов (**Device Tolerance**) 1%. Оба резистора имеют допуск группы (**Lot Tolerance**) 4% и одинаковый **Lot Tracking Number**. При каждом запуске процесса моделирования методом Monte Carlo резисторам сначала назначается одинаковое групповое отклонение от номинального значения в диапазоне $\pm 4\%$, а затем от полученного значения каждому резистору назначается допуск элемента в диапазоне $\pm 1\%$. В итоге суммарный допуск равен 5% (1%+4%). Тем не менее, из-за одинакового числа **Lot Tracking Number** в течение одного сеанса моделирования значения для каждого резистора не могут отличаться более чем на +/- 1% от их номинального значения или максимум на 2% друг от друга.

Запуск анализа методом Monte Carlo

Для запуска анализа методом Monte Carlo необходимо:

1. Установить параметры анализа методом Monte Carlo, как описано выше.
2. Включить опцию **Monte Carlo Analysis** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
3. Результатом выполнения анализа методом Monte Carlo может стать обработка большого количества данных. Для ограничения количества обрабатываемых данных необходимо в списке **Collect Data For** на вкладке **General** диалогового окна **Analyses Setup** установить значение **Active Variables**. Это позволит, при проведении анализа, обрабатывать только те переменные, которые в настоящий момент перечислены в окне **Active Variables**. На рисунке изображенном выше показано, что при расчете будет обрабатываться только переменная **OUTPUT**.
4. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается, и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информации обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они появляются по мере расчета.

В зависимости от того, какой тип анализа (**AC**, **DC**, или **Transient**) был выбран, программа моделирования отображает результаты в соответствующем окне, где имеется возможность просматривать рассчитанные кривые и уточнять отдельные их значения. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Автоматическое изменение параметров элементов (Parameter Sweep)

Функция **Parameter Sweep** позволяет производить параметрический анализ — автоматическое изменение значений параметров элементов с определенным шагом в заданном диапазоне (рис. 4.12). Программа моделирования производит многократный запуск выбранного типа анализа (**AC**, **DC**, или **Transient**). С помощью функции **Parameter Sweep** можно изменять параметры только базовых компонентов и моделей. Параметры подсхем изменены быть не могут.

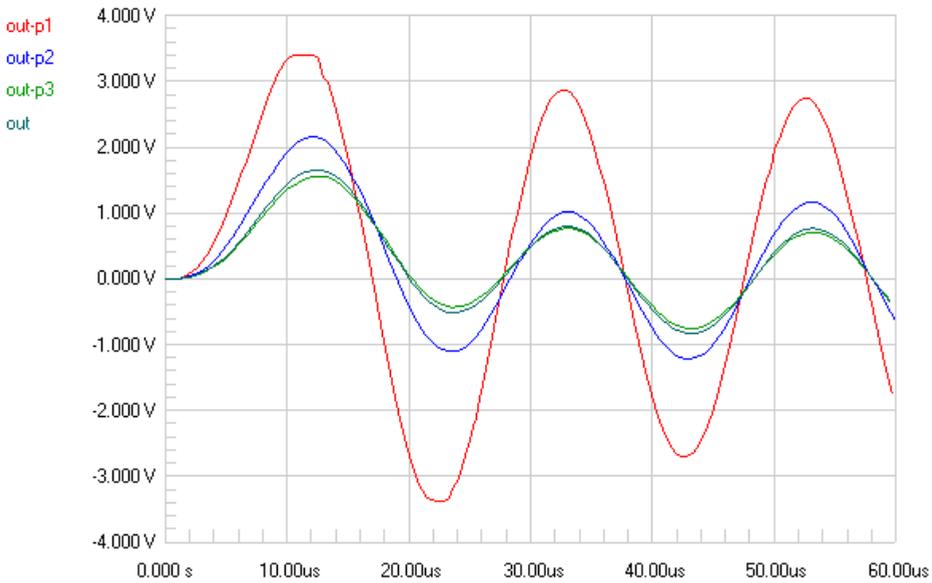


Рис. 4.12. Параметрический анализ.

Кроме этого, имеется возможность определить второй изменяемый параметр. В случае, когда этот параметр задан, изменение основного (**Primary**) параметра производится для каждого значения второго (**Secondary**) параметра. Например, если шаг изменения установлен таким образом, что будет 3 значения основного параметра и 3 значения второго параметром, то элемент будет иметь следующие наборы параметров: **P1** и **S1**, **P2** и **S1**, **P3** и **S1**, затем **P1** и **S2**, **P2** и **S2**, **P3** и **S2** и так далее.

Установка режима автоматического изменения параметров (Parameter Sweep)

Установка режима изменения параметров производится на вкладке **Parameter Sweep** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup** (рис. 4.13).

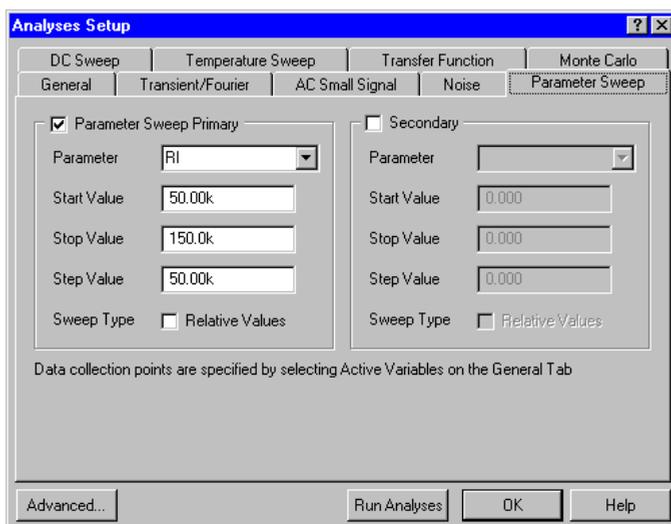


Рис. 4.13. Настройка параметрического анализа.

При установке режима изменения параметров элементов необходимо задать следующие данные: **Parameter**, **Start Value**, **Stop Value**, **Step Value**. В качестве изменяемого параметра может использоваться обозначение компонента, например **C2**, или обозначение компонента с дополнительным параметром, указываемым в скобках, например, **U5[tp_val]** (таблица 4.2).

Таблица 4.2

Пример	Изменяемый параметр
RF	Резистор с обозначением RF
Q3[bf]	Коэффициент передачи по току транзистора Q3
R3[r]	Сопротивление потенциометра R3
option[temp]	Температура
U5[tp_val]	Задержка распространения цифрового устройства U5

Обычно, для задания изменений температуры в процессе моделирования используется вкладка **Temperature Sweep**, однако, также существует возможность установки температуры на вкладке **Parameter Sweep**. Это бывает полезным при необходимости изменения температуры одновременно с изменением основного или дополнительного параметра.

Когда использовать опцию Use Relative Values?

Если включить опцию использования относительных значений (**Use Relative Values**) на вкладке **Temperature Sweep**, то значения вводимые в поля **Start Value**, **Stop Value**, **Step Value** будут использоваться совместно к уже заданным параметрам или к параметрам со значениями по умолчанию. Для примера, рассмотрим функцию изменения параметра при следующих условиях:

1. Изменяемый параметр — резистор с номиналом 1 кОм.
2. Значения полей **Start Value**, **Stop Value**, **Step Value** соответственно -50, 50, 20.
3. Опция **Use Relative Values** включена.

При запусках процесса моделирования резистор будет принимать следующие значения: 950, 970, 990, 1010, 1030, и 1050.

Запуск режима автоматического изменения параметров элементов (Parameter Sweep)

Для запуска режима автоматического изменения параметра элемента необходимо:

1. В процессе расчета в режиме **Parameter Sweep** может производиться обработка большого количества данных. Для ограничения количества обрабатываемых данных необходимо в списке **Collect Data For** на вкладке **General** диалогового окна **Analyses Setup** установить значение **Active Variables**. Это позволит, при проведении анализа, обрабатывать только те переменные, которые в настоящий момент перечислены в окне **Active Variables**.
2. Установить параметры для режима **Parameter Sweep**, как описано выше.
3. Включить опцию **Parameter Sweep** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
4. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета.

В зависимости от того, какой тип анализа (**AC**, **DC**, или **Transient**) был выбран, программа моделирования отображает результаты в соответствующем окне, где имеется

возможность просматривать рассчитанные кривые и уточнять отдельные их значения. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Режим изменения температуры (Temperature Sweep)

Режим изменения температуры может использоваться в сочетании с частотным анализом в режиме малого сигнала, расчетом рабочих точек, а также при анализе переходных процессов. Схема анализируется в заданном диапазоне температур, при этом для каждого значения температуры строится свой набор кривых.

Установка параметров режима изменения температуры (Temperature Sweep)

Установка параметров режима изменения температуры производится на вкладке **Temperature Sweep** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup** (рис. 4.14).

Установка одного или более стандартных типов анализа происходит таким образом, что каждый вид моделирования производится для указанной температуры.

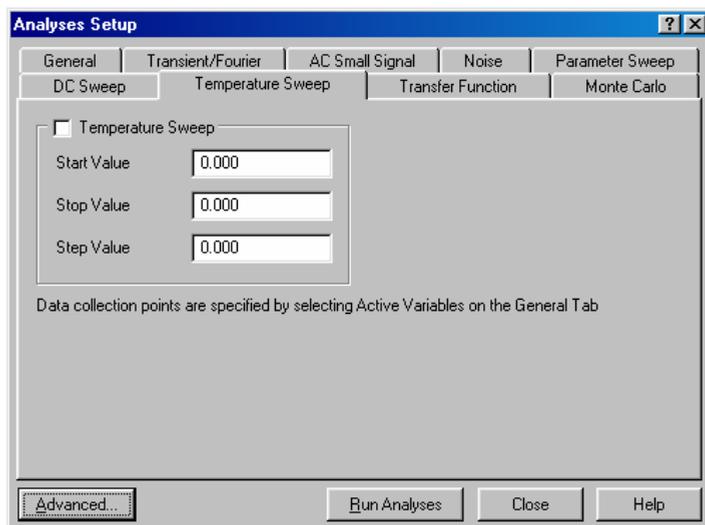


Рис. 4.14. Настройка режима изменения температуры.

Запуск режима изменения температуры (Temperature Sweep)

Для запуска режима изменения параметра необходимо:

1. В процессе функционирования в режиме **Temperature Sweep** может производиться обработка большого количества данных. Для ограничения количества обрабатываемых данных необходимо в списке **Collect Data For** на вкладке **General** диалогового окна **Analyses Setup** установить значение **Active Variables**. Это позволит, при проведении анализа, обрабатывать только те переменные, которые в настоящий момент перечислены в окне **Active Variables**.
2. Установить параметры для режима **Temperature Sweep** на вкладке **Temperature Sweep** диалогового окна **Analyses Setup**.
3. Включить опцию **Temperature Sweep** на вкладке **General** диалогового окна **Analyses Setup**.
4. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета.

В зависимости от того, какой тип анализа (**AC**, **DC**, или **Transient**) был выбран, программа моделирования отображает результаты в соответствующем окне, где имеется возможность просматривать рассчитанные кривые и уточнять отдельные их значения. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Анализ Фурье (Fourier Analysis)

Программа моделирования может производить анализ Фурье, основываясь на данных, получаемых в последнем такте анализа переходных процессов. Например, если основная частота сигнала равна 1 кГц, то данные из последнего 1 мс такта могут быть использованы для анализа Фурье.

Установка параметров для анализа Фурье

Установка параметров для анализа Фурье производится на вкладке **Transient/Fourier** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup**. Необходимо установить режим моделирования переходных процессов для выполнения анализа Фурье. Кроме этого, нужно задать основную частоту анализа и количество требуемых гармоник.

Запуск анализа Фурье

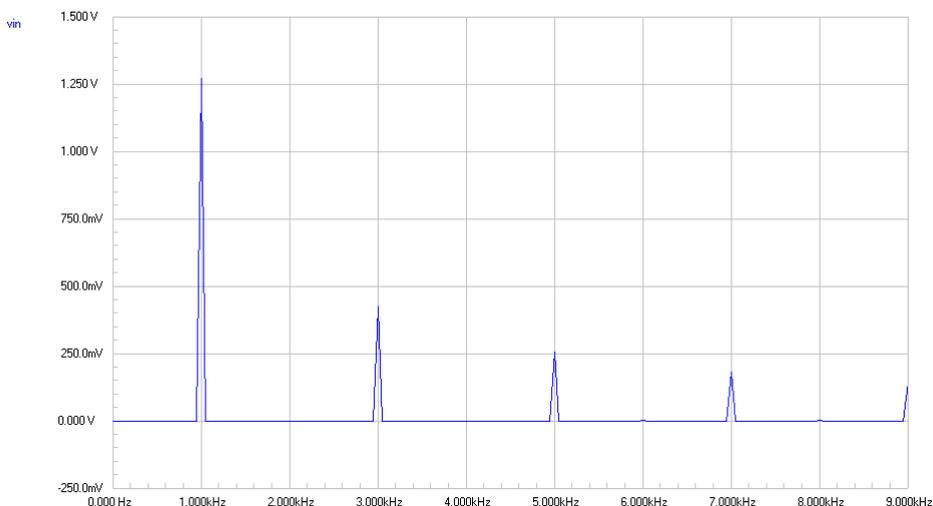


Рис. 4.15. При Фурье анализе рассчитывается спектр сигнала в заданном узле схемы.

Для запуска анализа Фурье необходимо:

1. Установить параметры анализа Фурье, как описано выше.
2. Включить опцию **Transient/Fourier Analysis** на вкладке **General** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup**.
3. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета. Более подробно информация о величине и фазе каждой гармонике представлена в файле с расширением **.SIM**.

Пользователь имеет возможность просматривать рассчитанные кривые Фурье и уточнять отдельные их значения. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Анализ передаточных функций (Transfer Function Analysis)

В процессе анализа передаточных функций происходит вычисление входного сопротивления по постоянному току, выходного сопротивления по постоянному току, и усиления по постоянному току.

Установка параметров для анализа передаточных функций

Установка параметров для анализа передаточных функций производится на вкладке **Transfer Function** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup**. Необходимо выбрать опорный источник сигнала и задать узел, относительно которого будут производиться вычисления.

Запуск анализа передаточных функций

Для запуска режима изменения параметра необходимо:

1. Установить параметры для режима **Transfer Function**, как описано выше.
2. Включить опцию **Transfer Function** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
3. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета.

Пользователь может просмотреть значения входного сопротивления по постоянному току, выходного сопротивления по постоянному току, и усиления по постоянному току для каждого узла схемы на вкладке **Transfer Function** окна моделирования. Более подробно работа с этим окном описана в разделе *Окно просмотра результатов расчета*.

Анализ шумов (Noise Analysis)

Анализ шумов позволяет измерять шумовые характеристики схемы путем определения шумов резисторов и полупроводниковых устройств. Программа моделирования строит кривую спектральной плотности шума, на которой шум измеряется в $V^2/Гц$. Конденсаторы, катушки индуктивности, и управляемые источники постоянного напряжения считаются идеальными, не вносящими дополнительных шумов в схему. Программа моделирования позволяет проводить следующие измерения шумовых характеристик:

Выходной шум

Позволяет измерять уровень шума на выходе схемы.

Входной шум

Определяет уровень входного шума, являющегося причиной рассчитанного уровня шума на выходе схемы. Например, если выходной шум равен 10p и схема имеет усиление 10, это означает, что шум на входе схемы должен быть равен 1p. Таким образом, эквивалентный входной шум будет равен 1p.

Шумовая характеристика компонентов

Позволяет определить шум, который вносит каждый компонент в выходной шум. Общий выходной шум равен сумме шумов резисторов и полупроводниковых устройств. Каждый из компонентов добавляет определенное количество шума, которое умножается на усиление, определяемое от позиции компонента до выхода схемы. Таким образом, один и тот же компонент может давать различные шумовые добавки к выходному шуму, в зависимости от его места расположения в схеме.

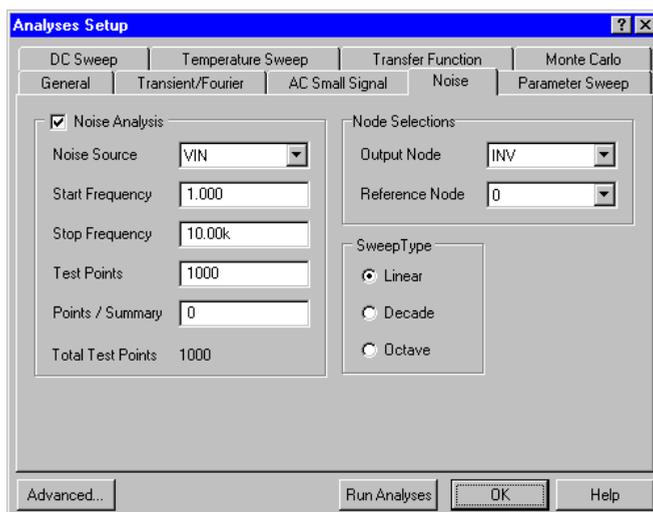


Рис. 4.16. Настройка параметров шумового анализа.

Установка параметров для анализа шумов

Установка параметров для анализа шумов производится на вкладке **Noise** диалогового окна **Analyses Setup**, которое вызывается с помощью команды меню **Simulate » Analyses Setup** (рис. 4.16).

Необходимо выбрать источник опорного сигнала, ввести начальную и конечную частоты, а также количество расчетных точек, в которых будут производиться вычисления. Распределение расчетных точек в частотном диапазоне определяется выбранной опцией **Sweep Type**. Ввод нулевого значения в поле **Points Summary** позволяет измерять только входной и выходной шум, задание единицы позволяет определять шумы каждого компонента в схеме.

Запуск анализа шумов

Для запуска анализа шумов необходимо:

1. Установить параметры для анализа шумов, как описано выше.
2. Включить опцию **Noise Analysis** на вкладке **General** диалогового окна **Analyses Setup**, вызываемого командой меню **Simulate » Analyses Setup**.
3. После установки этой опции можно запустить процесс моделирования нажатием кнопки **Run Analyses** в нижней части диалогового окна или командой меню **Simulate » Run**.

Ход процесса моделирования отражается в строке состояния. Если во время создания списка соединений обнаруживается ошибка, процесс прерывается и появляется окно с сообщением, предлагающим просмотреть файл ошибок. Просмотр этого файла дает возможность исправить ошибки. За более подробной информацией обратитесь к разделу *Обработка ошибок, возникающих при моделировании*.

Как только процесс создания списка соединений завершается, появляется окно отображения результатов моделирования, где они отображаются по мере расчета.

Пользователь может просмотреть результаты анализа шумов в окне отображения результатов моделирования. Кривые входных и выходных шумов помечаются соответственно **NI (input noise)** и **NO (output noise)**.

Расчет комплексных сопротивлений (Impedance Plot Analysis)

Расчет комплексных сопротивлений показывает импеданс схемы в точках включения какого-либо источника. Этот тип анализа не имеет отдельного диалогового окна установки параметров, и обычно отображается в окне **AC Analysis**.

Установка параметров расчета комплексных сопротивлений

Что бы включить результаты расчета комплексных сопротивлений в набор выводимых данных, можно воспользоваться любой из двух опций содержащихся в списке **Collect Data For** на вкладке **General** диалогового окна **Analyses Setup**:

- **Node Voltages, Supply and Device Currents**

- **Node Voltages, Supply Currents, Device Currents and Powers**

Поместите интересующий объект в список доступных переменных (**Available Variables**) и список активных переменных (**Active Variables**). Источник сигнала с обозначением **VIN** будет отображаться в списке как **@VIN(z)**.

Запуск расчета комплексных сопротивлений

При запуске расчета комплексных сопротивлений они будут отображаться в окне просмотра результатов анализа. Это особенный интерес представляет зависимость импеданса схемы от частоты, изображенная совместно с частотными ее характеристиками в режиме малого сигнала. Импеданс вычисляется как отношение напряжения на положительном выводе источника и тока, вытекающего из этого вывода.

Имеется возможность рассчитать выходной импеданс схемы, для чего необходимо:

1. Удалить источник сигнала со входа.
2. Заземлить все входы схемы, с которыми этот источник был соединен.
3. Удалить все нагрузочные элементы, присоединенные к схеме.
4. Присоединить к выходу схемы положительный вывод источника сигнала, отрицательный вывод заземлите.
5. Установить переменные (**Variables**) как описано выше.
6. Запустить процесс моделирования.

Для отображения на графике модуля комплексного сопротивления нужно установить по оси Y величину **Magnitude**, для чего в режиме **Single cell** посредством нажатия правой кнопки мыши в этом окне и выбора команды **Scaling** из появившегося всплывающего меню вызывается диалоговое окно **Scaling Options**, где делаются необходимые установки.

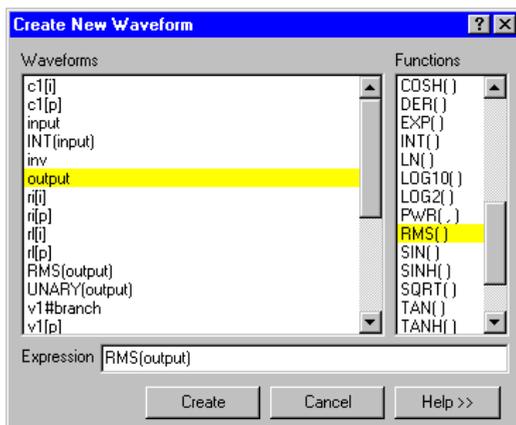


Рис. 4.17. При математических действиях над сигналами могут быть использованы различные функции.

Математические действия с рассчитанными сигналами

При моделировании электронных схем иногда требуется произвести некоторые математические действия с рассчитанными сигналами и посмотреть результат. Данная функция интегрирована в окно просмотра результатов расчета и позволяет строить математические выражения с использованием любого из сигналов или характеристик.

Математические функции задаются в диалоговом окне **Create New Waveform**, вызываемом нажатием кнопки **New** на панели управления программой моделирования. Здесь присутствует список доступных сигналов **Waveforms**, список доступных функций **Functions** и поле построения выражения **Expression** (рис. 4.17). Выражение может быть записано как непосредственно, так и выбором нужных сигналов и функций из соответствующих списков.

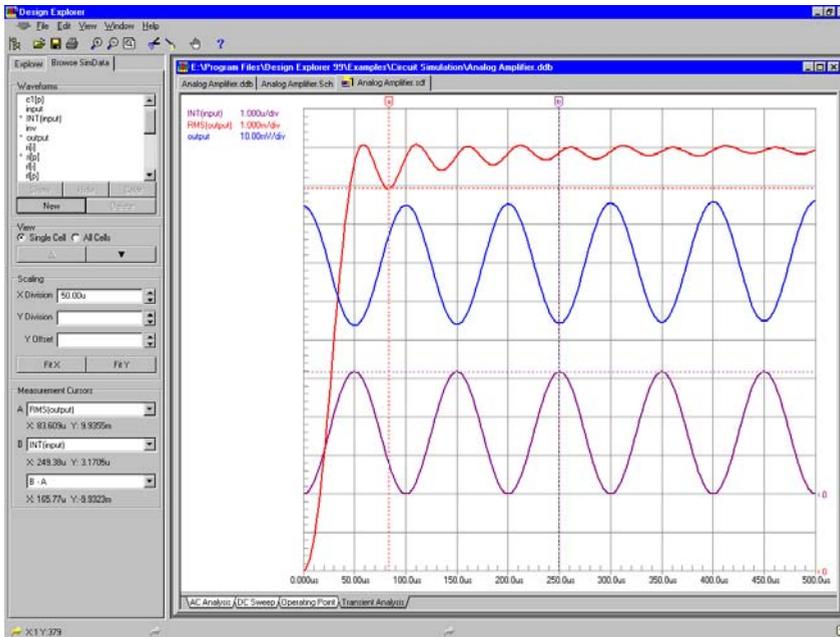


Рис. 4.18. Результаты математических преобразований отображаются на графиках также, как и обычные сигналы и характеристики.

Используемые функции и операторы

Функции и операторы, использующиеся для записи математических выражений для преобразования сигналов, представлены в таблице 4.3.

Таблица 4.3

Оператор или функция	Описание
+	Сложение
-	Вычитание
*	Умножение
/	Деление
^	Возведение в степень, работает аналогично функции PWR(,).
()	Приоритет вычисления
ABS()	Абсолютное значение. ABS(x) возвращает значение x .
ACOS()	Арккосинус
ACOSH()	Гиперболический арккосинус
ASIN()	Арксинус
ASINH()	Гиперболический арксинус
ATAN()	Арктангенс
ATANH()	Гиперболический арктангенс
AVG()	Функция осреднения, возвращает среднее значение сигнала
BOOL(,)	Булева функция. Запись BOOL(сигнал, порог) генерирует единицу, если сигнал выше порогового значения, и ноль, если ниже этого значения.
COS()	Косинус
COSH()	Гиперболический косинус

Таблица 4.3. Продолжение

Оператор или функция	Описание
DER()	Производная dx/dt , возвращает наклон отрезков кривой между точками данных.
EXP()	Экспоненциальная функция
INT()	Интегральная функция
LN()	Натуральный логарифм
LOG10()	Логарифм по основанию 10
LOG2()	Логарифм по основанию 2
PWR(,)	Возведение в степень. Работает аналогично оператору ^
RMS()	Среднеквадратичное значение
SIN()	Синус
SINH()	Гиперболический синус
SQRT()	Корень квадратный
TAN()	Тангенс
TANH()	Гиперболический тангенс
UNARY()	Инверсия знака. Запись UNARY(x) возвращает значение -x.
URAMP()	Функция скачка. Возвращает значение 0, если аргумент x меньше 0, и x, если больше или равно 0.
USTEP()	Функция единичного скачка (Хевисайда). Возвращает значение 1, если аргумент меньше 0, и 0, если аргумент больше или равен 0.

Включение дополнительной SPICE информации в список соединений

Дополнительная SPICE информация для моделирования может быть включена в список соединений путем введения в схему текстового окна. Первая строка текста должна содержать директиву **.NSX**. Все остальные строки текста будут включены в список соединений в том виде, в котором они были введены в текстовое окно.

Пример использования текстового окна для определения индуктивной связи

Представленный на рисунке 4.19 пример демонстрирует использование текстового окна **Text Frame** для задания индукционной связи между двумя катушками индуктивности. В этом примере **K1** представляет собой трансформатор с индуктивной связью (название может быть произвольным, начинающимся с буквы **K**), **L1** и **L2** — обозначения катушек индуктивности, а **0.5** — коэффициент связи.

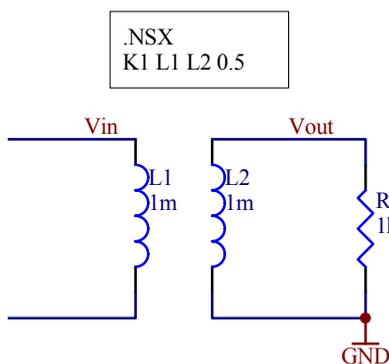


Рис. 4.19. Добавление на схему описания индуктивной связи.

Создание списка соединений и запуск процесса моделирования

При запуске процесса моделирования из окна редактора принципиальных схем информация передается в модуль моделирования в форме списка соединений. Этот список соединений создается в виде временного файла, который автоматически удаляется по завершению процесса моделирования.

Существует возможность выполнения моделирования непосредственно из списка соединений. Для этого список соединений должен быть открыт в окне редактора (для каждого открытого в данный момент документа имеется собственная закладка с именем файла с расширением **.NSX**). Вышесказанное относится не только к активному

документу. Для создания, просмотра и редактирования списка соединений используется команда меню **Simulate » Create Spice Netlist**.

Примечание: Список соединений, не открытый и хранящийся в базе данных проекта, для моделирования не используется. В случае, если список соединений в данный момент не открыт, то создается временный список соединений, который и используется при моделировании. Если список соединений в данный момент уже открыт, то для моделирования используется именно он.

Установка начальных условий

В некоторых случаях, например, при проектировании неустойчивых и бистабильных схем, перед выполнением моделирования может потребоваться предустановка значений напряжения в узлах схемы. Программа моделирования содержит для этого несколько инструментов, включая модули установки параметров узла (**Nodeset**) и начальных условий (**Initial Condition**), а также опцию установки начальных условий **Use Initial Condition**. Модули установки параметров узла и начальных условий хранятся в библиотеке символов моделирования **Symbols.lib**.

Модуль .NS (установка параметров узла)

NS1
12

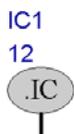


Модуль установки параметров узла используется для задания начального напряжения в узле схемы во время предварительного прохода расчета рабочих точек. После предварительного прохода ограничения снимаются и итерации продолжают до получения правильного значения смещения. При размещении модуля необходимо задать атрибуты, приведенные в таблице 4.4.

Таблица 4.4

Атрибут	Значение
Designator	Каждый модуль установки параметров узла должен иметь уникальное позиционное обозначение
Part Type	Амплитуда напряжения в узле (например, 12)

Модуль .IC (начальные условия)



Модуль задания начальных условий используется для установки временных начальных условий. Способ, которым программа моделирования определяет начальные условия, зависит от .IC модулей, и устанавливается опцией **Use initial Conditions**, расположенной на вкладке **Transient/Fourier** диалогового окна **Analyses Setup**.

Если опция **Use initial Conditions** не включена на этапе расчета рабочих точек, напряжение в узле определяется значением, задаваемым .IC модулем. Во время последующего анализа переходных процессов это ограничение снимается. Этот метод является предпочтительным, так как позволяет программе моделирования SPICE получить правильное решение по постоянному току. При размещении модуля необходимо задать атрибуты, приведенные в таблице 4.5.

Таблица 4.5

Атрибут	Значение
Designator	Каждый модуль начальных условий должен иметь уникальное позиционное обозначение
Part Type	Амплитуда напряжения в узле (например, 5)

Использование опции установки начальных условий

В случае установки опции **Use Initial Conditions** на вкладке **Transient/Fourier** диалогового окна **Analyses Setup** расчет рабочих точек не производится. Вместо этого напряжения в узлах определяются .IC модулями, которые используются при вычислении начальных условий для конденсаторов, диодов, биполярных, полевых и МОП транзисторов. Поскольку расчет рабочих точек не производится, необходимо установить соответствующие значения напряжения для всех узлов. Подробности по установке опции использования начальных условий приведены в подразделе *Анализ переходных процессов (Transient Analysis)* раздела *Подготовка и проведение моделирования*.

Окно просмотра результатов расчета

Программа моделирования отображает данные и результаты моделирования в многостраничном окне просмотра результатов моделирования, которое позволяет легко и быстро производить визуальный анализ полученных результатов (рис. 4.20).

Результаты каждого назначенного типа анализа отображаются на отдельной вкладке этого окна. Подробная информация по конфигурированию каждого из типов анализа приведена в разделе *Подготовка и проведение моделирования*.

Способ задания узлов схемы, для которых информация будет отображаться в окне просмотра результатов расчета, изложен в пункте *Сбор, отображение и сохранение результатов моделирования* раздела *Подготовка и проведение моделирования*.

Окно просмотра результатов моделирования функционирует аналогично обычному осциллографу. Команды масштабирования расположенные на панели управления позволяют выборочно просматривать нужные участки изображения. Здесь также имеются инструменты, позволяющие производить измерения параметров сигналов непосредственно в окне.

Отображение формы сигнала

Результаты расчета отображаются по мере выполнения моделирования. По завершению процесса моделирования, для просмотра результатов каждого из назначенных типов анализа необходимо щелкнуть на соответствующей закладке в верхней части окна. Результаты расчета рабочих точек представляются в виде списка значений напряжения, тока и мощности для узлов или элементов схемы.

Обратите внимание, что отображение результатов расчета рабочих точек зависит от значения параметра **Collect Data** устанавливаемого на вкладке **General** диалогового окна **Analyses Setup**.

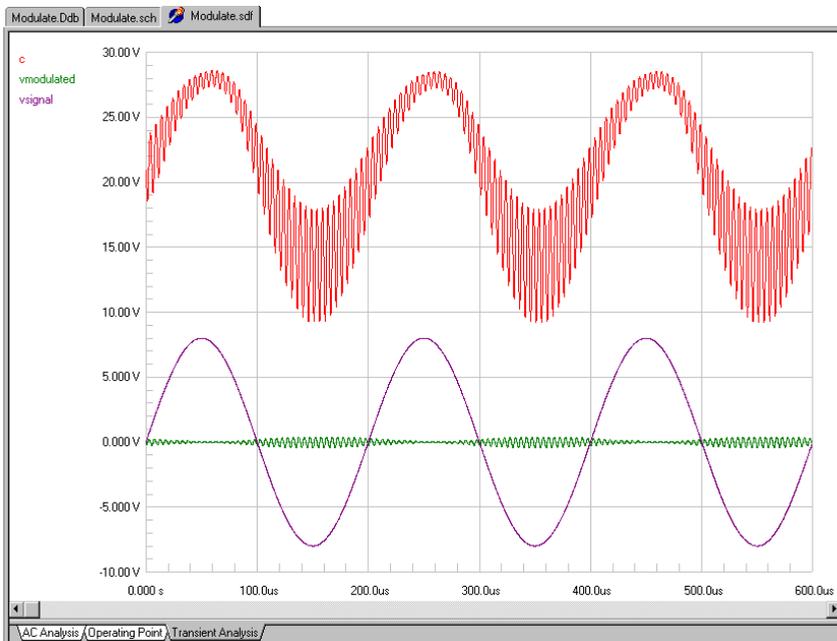


Рис. 4.20. Окно просмотра результатов расчета.

Масштабирование кривых

При первоначальном отображении результаты расчета масштабируются автоматически. По оси Y все результаты масштабируются пропорционально таким образом, чтобы все сигналы оставались видимыми, а самый большой сигнал занимал почти все окно. Масштабирование по оси X будет зависеть от установок каждого из назначенных типов анализа. Например, для анализа переходных процессов общая ширина окна по оси X определяется разностью значений **Stop Time** и **Start Time**.

Для изменения размера или расположения сигналов воспользуйтесь функциями настройки масштаба (рис. 4.22), расположенными на панели управления (для отображения панели нажмите **View » Panel**).

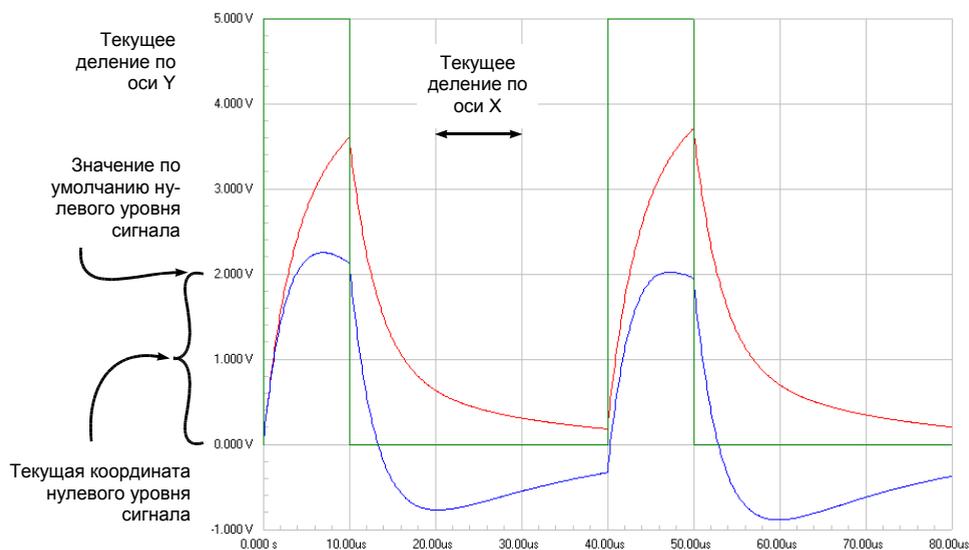


Рис. 4.21. Пример масштабирования кривых.

Масштабирование по оси X

Для растягивания или сжатия рассчитанных кривых в горизонтальном направлении необходимо изменить на панели управления параметр **X Division Size**. Изменение масштаба возможно как вводом нового значения, так и с помощью стрелок, находящихся рядом с окном ввода масштабного коэффициента.

Масштабирование по оси Y

Для растягивания или сжатия рассчитанных кривых в вертикальном направлении необходимо изменить

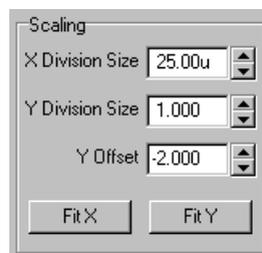


Рис. 4.22. Органы управления масштабированием.

на панели параметр **Y Division Size**. Обратите внимание, что на положение масштабируемого сигнала влияет текущее значение параметра **Y Offset**. Если при изменении параметра **Y Division Size** сигнал исчезнет из области видимости, необходимо временно установить параметр **Y Offset** равным 0.

Настройка параметра Y Offset

Параметр **Y Offset** представляет собой координату нулевого уровня сигнала, которую можно перемещать вдоль оси Y. По умолчанию нулевой уровень располагается в центре оси Y. Отрицательное значение **Y Offset** показывает, что уровень нуля смещается вниз, а положительное значение **Y Offset** показывает, что уровень нуля смещается вверх. Для настройки этого параметра необходимо ввести новое значение в поле **Y Offset** или воспользоваться стрелками, расположенными рядом с окном ввода.

Сброс масштабирования кривой

Процедура сброса масштабирования одинакова для всех сигналов: сначала нужно убедиться, что ни один из сигналов не выбран (то есть отсутствует точка рядом с его именем), а затем щелкнуть на кнопке **Fit Y** расположенной на панели.

Настройка параметров отображения отдельного сигнала

Перед настройкой масштаба отдельного сигнала его необходимо "выбрать". Выбор сигнала осуществляется щелчком на его имени в списке расположенном сбоку от сигналов (рис. 4.23).

Обратите внимание на изменение цвета кривой при масштабировании показывающее, что масштабирование касается только выбранного сигнала.

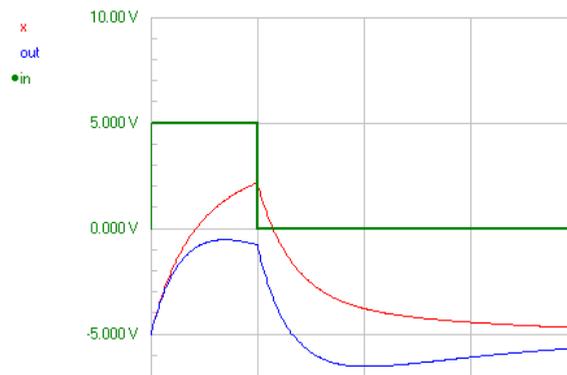


Рис. 4.24. Результат изменения масштаба.

Для изменения масштаба выбранного сигнала используются инструменты масштабирования. Допускается как изменение параметра **Y Division Size** (размер деления по оси

Y), так и параметра **Y Offset** (смещение по оси Y). В примере на рисунке 4.24 размер деления по оси изменяется от 2.5 В на деление до 5 В на деление, смещение по оси Y сдвигает сигнал **In** вверх.

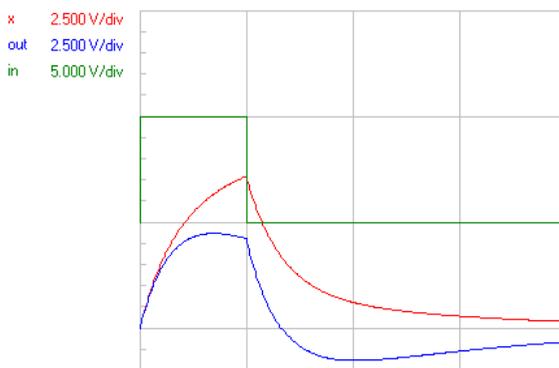


Рис. 4.25. Отмена выбора сигнала.

Отмена выбора сигнала **In** производится повторным щелчком на его имени. Обратите внимание на исчезновение шкалы вдоль оси Y, вместо нее возле каждого имени сигнала показывается его масштаб — это происходит только в случае различия масштабов (рис. 4.25).

Просмотр сигналов на отдельных графиках

Для просмотра сигналов на отдельных графиках нужно воспользоваться командой **All Cells** расположенной на панели управления. Эту возможность можно использовать при необходимости отдельного просмотра цифровых сигналов.

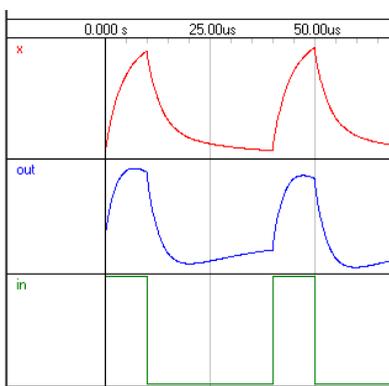


Рис. 4.26. Просмотр сигналов на отдельных графиках.

Отображение нескольких сигналов на одном графике

По умолчанию при первом проходе моделирования отображение рассчитанных сигналов осуществляется на отдельных графиках. Щелкнув левой кнопкой мыши на нужном графике и удерживая ее можно перетащить график на другой, после чего при переключении режима отображения в состояние **Single Cell** они будут отображаться вместе (рис. 4.27).

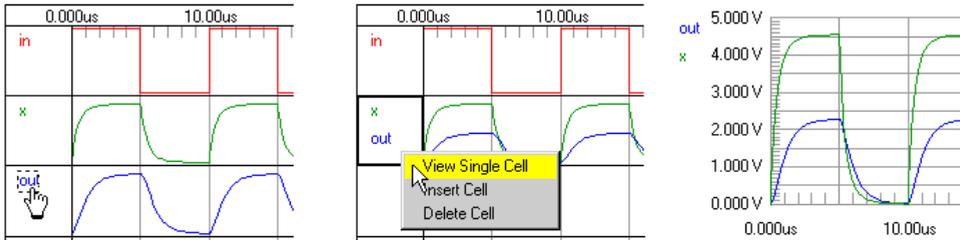


Рис. 4.27. Для того, чтобы объединить два графика достаточно просто перетащить один на другой.

- ◆ Переключение между режимами **All Cells** (отдельные графики) и **Single Cell** (один график) производится щелчком правой кнопки мыши на нужной кривой.

Одновременное отображение кривых, имеющих разный масштаб по оси Y

Иногда бывает необходимо отобразить на одном графике две кривые, имеющие разный масштаб по оси Y, например, амплитудно- и фазочастотные характеристики или ХГВЗ (рис. 4.28).

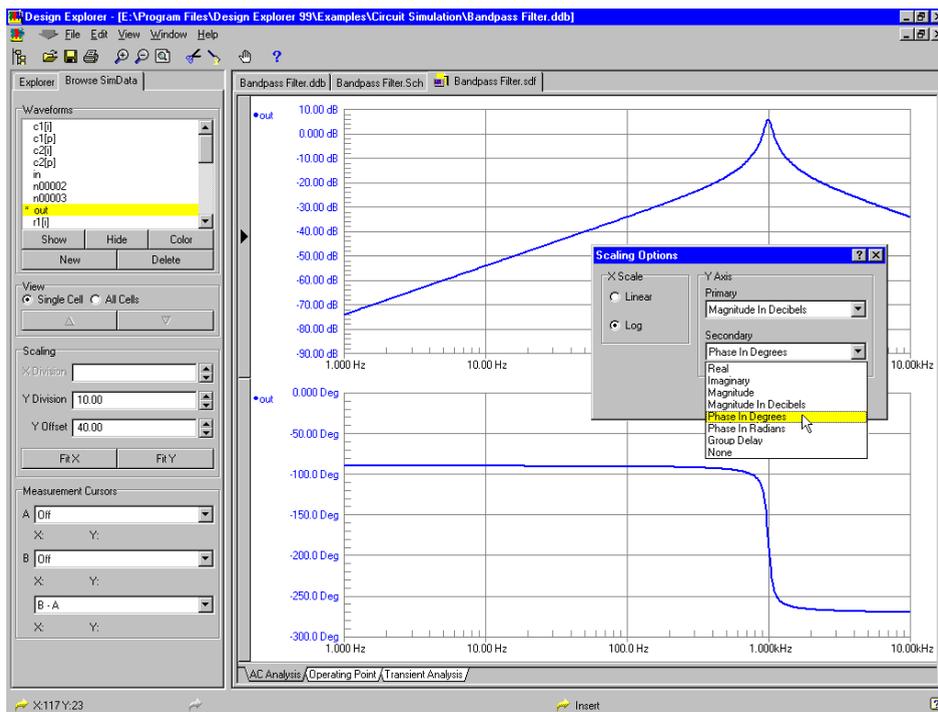


Рис. 4.28. Одновременный просмотр частотной и фазовой характеристик

Для этого нужно на одном из графиков выполнить щелчок правой кнопкой мыши и в появившемся всплывающем меню выбрать команду **Scaling**. Второй масштаб по оси Y задается в выпадающем списке **Secondary**.

Измерительный маркер в этом случае используется, как и на простом графике.

Изменение формата отображения результатов расчета

Результаты моделирования по переменному току могут отображаться в различных форматах. Здесь может быть использован линейный или логарифмический масштаб по оси X, значения модуля, амплитуды или фазы по оси Y. Для изменения формата

отображения данных во время просмотра результатов необходимо выбрать команду меню **Option** » **Scaling**. Кроме того, вызов диалогового окна **Scaling Option** осуществляется при выборе команды **Scaling** всплывающего меню, которое появляется при щелчке правой кнопкой мыши на окне просмотра результатов моделирования.

Отображение точек данных моделирования

При отсутствии уверенности в точности полученных результатов, например, когда они выглядят слишком изрезанными вместо того, чтобы быть достаточно гладкими, можно включить режим отображения рассчитанных точек для проверки того, что отсчеты вычислялись достаточно часто.

Для отображения точек данных моделирования нужно выполнить команду меню **Design** » **Options** и в появившемся диалоговом окне **Document Option** включить опцию **Show Data Points**. В каждой точке, для которой в ходе расчета были вычислены данные, будет отображаться маленький кружок.

Распознавание графиков при монохромной печати

Окно просмотра результатов моделирования содержит функцию, которая облегчает распознавание кривых на графике при монохромной печати результатов. Если выбрать команду меню **Design** » **Option** и включить опцию **Show Data Points**, к каждой отображаемой кривой добавляются идентификаторы. Если сигналы отображаются на отдельных графиках, то для всех сигналов используется идентификатор в виде небольшого квадрата. В случае, если на одном графике отображается два или более сигналов, то для каждой кривой используются идентификаторы разной формы.

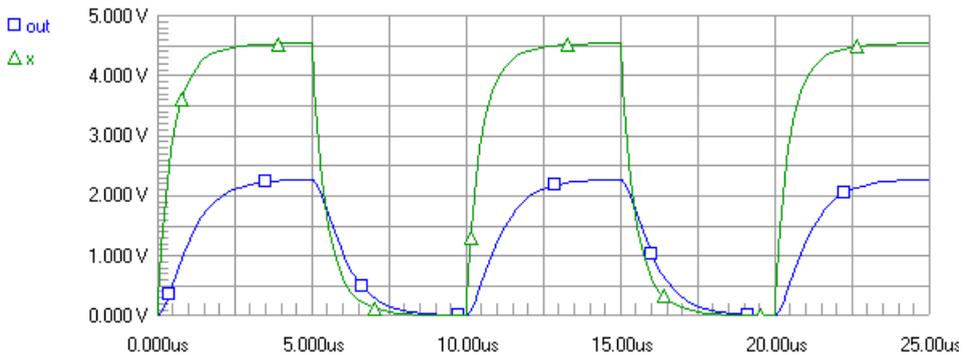


Рис. 4.29. При монохромной печати к кривым рекомендуется добавлять специальные графические идентификаторы

Использование измерительных маркеров

Для проведения непосредственных измерений сигналов имеется возможность использовать два измерительных маркера. Например, на кривых, полученных в результате выполнения анализа переходных процессов, можно измерить время и уровень сигнала, а на частотных характеристиках, полученных в результате малосигнального анализа, можно оценить полосу пропускания устройства по уровню 3 дБ. Имеется возможность вычисления показаний двух маркеров, которые могут располагаться как на одном, так и на двух разных графиках.

Органы управления маркерами располагаются на панели управления окна просмотра (рис. 4.30).

Каждый из двух измерительных маркеров может быть назначен для любой из рассчитанных кривых. Для этого необходимо:

1. В выпадающем списке выбрать сигнал, которому требуется назначить маркер.
2. В верхней части окна появится маленький ярлык с буквой, обозначающий имя маркера.
3. В поле **Measurements Cursor** на панели управления появятся значения X и Y для текущей позиции маркера.

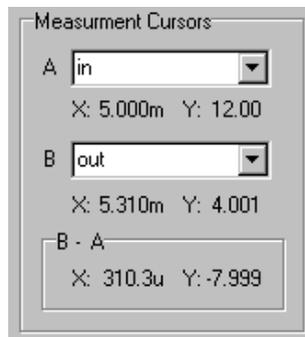


Рис. 4.30. Органы управления маркерами.

Способ отображения результатов нескольких проходов моделирования

При некоторых типах анализа, таких как статистический анализ методом Монте-Карло, а также параметрическом и температурном анализе, производится несколько последовательных проходов моделирования, причем на каждом проходе изменяется один или более параметров схемы.

Каждый проход идентифицируется буквой и цифрой, добавляемыми к имени сигнала. Буква обозначает тип многопроходного анализа: **m** — анализ методом Монте-Карло, **t** — температурный анализ, **p** — параметрический анализ, а цифра обозначает номер прохода. Буква и цифра не добавляются к метке результата моделирования, запуск которого происходил с номинальными значениями параметров. В таблице 4.6 представлено несколько примеров обозначения сигналов.

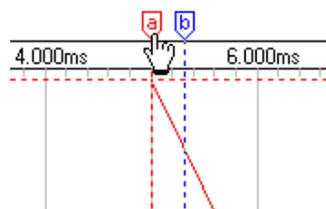


Рис. 4.31. Для перемещения измерительного маркера используется метка в верхней части окна.

Таблица 4.6

Обозначение	Интерпретация
output-p1	Напряжение в узле output, при первом запуске параметрического анализа
output-p2	Напряжение в узле output, при втором запуске параметрического анализа
output-p3	Напряжение в узле output, при третьем запуске параметрического анализа
output	Напряжение в узле output, при запуске с номинальными параметрами
output-m1	Напряжение в узле output, при первом запуске анализа методом Монте-Карло
output-t1	Напряжение в узле output, при первом запуске температурного анализа

Источники напряжения и тока

Перед тем, как осуществлять моделирование проекта необходимо включить в него соответствующие источники питания и входных сигналов.

Размещение компонентов источников

Источники сигналов можно разместить на схеме с помощью команды меню **Simulate » Sources** или напрямую из библиотеки символов моделирования, расположенной в библиотечной базе данных `Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`. Рядом с изображением каждого источника будет показано его библиотечное имя. Для установки атрибутов источника необходимо после размещения выполнить на нем двойной щелчок левой кнопкой мыши.

При задании параметров источника нет необходимости указывать единицы измерения (В или А). Если требуется, можно указать буквенный масштабный множитель (то есть meg, u, k и так далее). Описание разрешенных буквенных множителей приведено в разделе *Введение в моделирование*.

Источник постоянного напряжения или тока



Для питания схемы используется источник постоянного напряжения. После его размещения на схеме необходимо задать атрибуты, представленные в таблице 4.7.

Таблица 4.7

Атрибут	Значение
Designator	Задается требуемое имя шины питания (например, VDD)
Part Type	Амплитуда источника напряжения или тока (например, 12)
AC Magnitude (В или А)	Этот параметр устанавливается в случае, если данный источник планируется использовать в качестве сигнала при малосигнальном анализе по переменному току (типичное значение 1 В)
AC Phase (градусы)	Начальная фаза сигнала при проведении малосигнального анализа

Примечание: источник VSRC2 по умолчанию соединен с цепью GND.

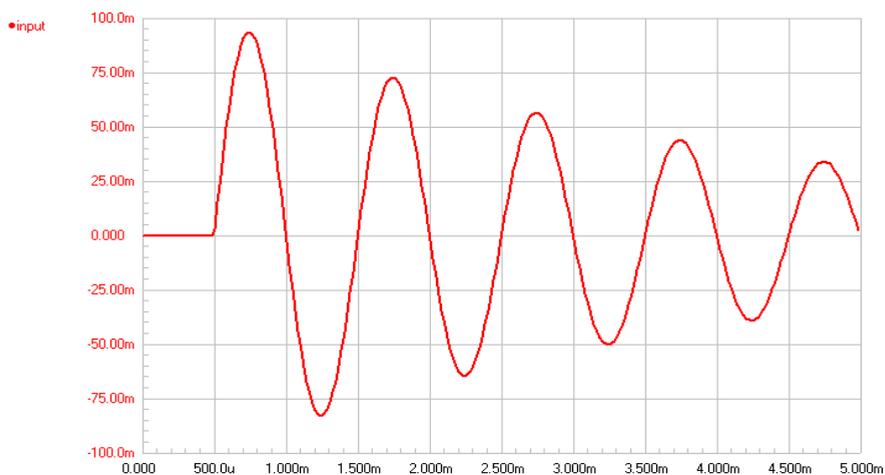
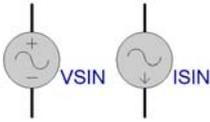


Рис. 4.32. Пример синусоидального сигнала.

Источник синусоидального сигнала



Данный источник используется для получения сигнала синусоидальной формы (рис. 4.32). После его размещения на схеме необходимо задать атрибуты, представленные в таблице 4.8.

Таблица 4.8

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, INPUT)
DC (В или А)	Смещение по постоянному току, используемое при расчете рабочих точек
AC (В или А)	Этот параметр устанавливается в случае, если планируется проводить малосигнальный анализ по переменному току (типичное значение 1 В)
AC Phase (градусы)	Начальная фаза источника напряжения или тока при проведении малосигнального анализа
Offset (В или А)	Постоянное смещение
Amplitude (В или А)	Пиковое значение амплитуды синусоидального сигнала (например, 100m)
Frequency (Гц)	Частота синусоидального напряжения или тока (например, 1000)
Delay (с)	Задержка перед началом сигнала (например, 500u)
Damping Factor (1/c)	Коэффициент затухания амплитуды синусоидального сигнала (например, 250)
Phase (градусы)	Сдвиг фазы синусоиды (в градусах) в нулевой момент времени (0)

Определение формы синусоидального сигнала

Сигнал, начинающийся в момент **Start Delay**, описывается приведенной ниже формулой, где t — текущий отсчет времени:

$$V(t_0 \text{ to } t_{SD}) = VO$$

$$V(t_{SD} \text{ to } t_{STOP}) = VO + VA \sin(2\pi F (t - SD)) e^{-(t - SD) \text{ THETA}}$$

DC Offset (VO)

Применяется для задания постоянного смещения генератора сигнала в В или А относительно отрицательного вывода, обычно соединенного с землей.

Peak Amplitude (VA)

Амплитуда выходного колебания, без учета постоянной составляющей, измеренная в В или А.

Frequency (F)

Частота выходного сигнала в Гц.

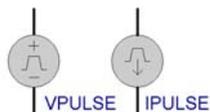
Start Delay (SD)

Фазовый сдвиг выходного сигнала, обеспечиваемый начальной задержкой синусоиды.

Damping Factor (THETA)

Положительное значение этого параметра дает экспоненциальное уменьшение амплитуды, отрицательное значение — экспоненциальное увеличение амплитуды.

Источник периодических импульсов



Данный источник используется для получения повторяющейся последовательности импульсов (рис. 4.33). После его размещения на схеме необходимо задать атрибуты, представленные в таблице 4.9.



Рис. 4.33. Пример периодической последовательности импульсов

Таблица 4.9

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (INPUT)
DC (В или А)	Смещение по постоянному току, используемое при расчете рабочих точек
AC (В или А)	Этот параметр устанавливается в случае, если планируется проводить малосигнальный анализ по переменному току (типовое значение 1 В)
AC Phase (градусы)	Фаза источника напряжения или тока при проведении малосигнального анализа
Initial (В или А)	Напряжение или ток в начальный момент времени (например, 0)
Pulsed (В или А)	Напряжение или ток в точке Time Delay + Rise Time (например, 5)
Time Delay (с)	Задержка перед изменением сигнала источника от начального значения
Rise Time (с)	Время нарастания от начального напряжения (Initial) до напряжения импульса (Pulsed), должно быть больше 0 (например, 4u)
Fall Time (с)	Время спада напряжения от напряжения импульса (Pulsed) до начального напряжения (Initial), должно быть больше 0 (например, 1u)
Pulse Width (с)	Время, в течение которого сохраняется напряжение импульса (Pulsed) (например, 0)
Period (с)	Время между началом первого и началом второго импульсов (например, 5u)

Задание формы импульса

Ниже приведено описание сигнала, где t — текущий отсчет времени. Промежуточные точки получаются методом линейной интерполяции:

$$\begin{aligned}
 V(t_0) &= VI \\
 V(t_{SD}) &= VI \\
 V(t_{SD} + t_{TR}) &= VP \\
 V(t_{SD} + t_{TR} + t_{PW}) &= VP \\
 V(t_{SD} + t_{TR} + t_{PW} + t_{TF}) &= VI \\
 V(t_{STOP}) &= VI
 \end{aligned}$$

Initial Amplitude (VI)

Начальная амплитуда выходного сигнала (измеренная в В или А), относительно отрицательного вывода, обычно соединенного с землей.

Pulse Amplitude (VP)

Максимальная амплитуда выходного колебания (в В или А).

Period (=1/freq)

Длительность одного полного периода выходного сигнала.

Pulse Width (PW)

Время, в течение которого выходной сигнал сохраняет уровень VP перед спадом до уровня VI.

Rise Time (TR)

Время нарастания от уровня VI до VP.

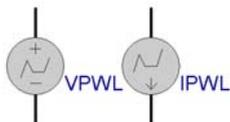
Fall Time (TF)

Время спада от уровня VP до VI.

Delay to Start (SD)

Время, в течение которого выходной сигнал сохраняет уровень VI перед первым нарастанием до уровня VP.

Источник кусочно-линейного сигнала



Данный источник используется для получения произвольного сигнала в виде набора значений напряжения или тока в различные моменты времени (рис. 4.34). После его размещения на схеме необходимо задать атрибуты, представленные в таблице 4.10.

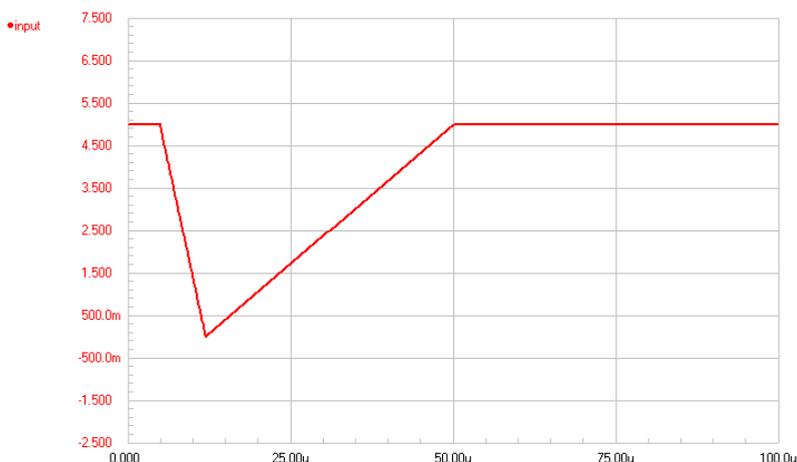


Рис. 4.34. Пример кусочно-линейного сигнала.

Таблица 4.10

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, INPUT)
DC (В или А)	Смещение по постоянному току, используемое при расчете рабочих точек
AC (V или A)	Этот параметр устанавливается в случае, если планируется проводить малосигнальный анализ по переменному току (типичное значение 1 В)
AC Phase (градусы)	Фаза источника напряжения или тока при проведении малосигнального анализа
Пары Time-Voltage или Time-Current	Значения напряжения или тока в каждый момент времени (например, 0u 5U 5u 5V 12u 0V 50u 5V 60u 5V)
File Name	Эта опция используется для связи кусочно-линейного сигнала, описанного во внешнем файле. Файл с расширением .PWL должен находиться в текущей директории.

Задание кусочно-линейного сигнала

Описание кусочно-линейного сигнала может быть сделано одним из двух способов:

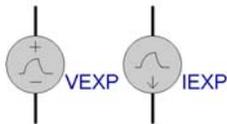
1. Существует возможность описания сигнала набором из точек (до восьми), которые вводятся непосредственно в поле **Time/Voltage** источника кусочно-линейного сигнала. Время, задаваемое для каждой последующей точки, должно быть больше, чем для предшествующей. Если в какой то точке это условие не соблюдается, то цикл будет закончен, и все последующие точки, начиная с этой, в него не войдут.
2. Сигнал может быть задан в виде ASCII файла содержащего любое количество точек. Значения должны вводиться парами. Каждая пара должна содержать амплитуду и время. Первым символом каждой строки данных должен быть знак плюса +, и каждая строка должна содержать не более 255 символов. Значения должны разделяться одним или более пробелами или знаками табуляции. Значения могут быть заданы как в простом, так и в экспоненциальном виде. В файл могут быть добавлены строки комментариев, первым символом такой строки должна быть звездочка *. Например:

```
* Random Noise Data
+ 0.00000e-3 0.6667 0.00781e-3 0.6372 0.01563e-3 -0.1177
+ 0.02344e-3 -0.6058 0.03125e-3 0.2386 0.03906e-3 -1.1258
+ 0.04688e-3 1.6164 0.05469e-3 -0.3136 0.06250e-3 -1.0934
+ 0.07031e-3 -0.1087 0.07813e-3 -0.1990 0.08594e-3 -1.1168
+ 0.09375e-3 1.4890 0.10156e-3 -0.2169 0.10938e-3 -1.4915
+ 0.11719e-3 1.4914 0.12500e-3 0.1486
```

Промежуточные точки определяются методом линейной интерполяции.

Примечание: Файл **.PWL** должен находиться в той же папке, что и моделируемая схема.

Источник сигнала экспоненциальной формы



Данный источник используется для получения сигнала экспоненциальной формы (рис. 4.35). После его размещения на схеме необходимо задать атрибуты, представленные в таблице 4.11.

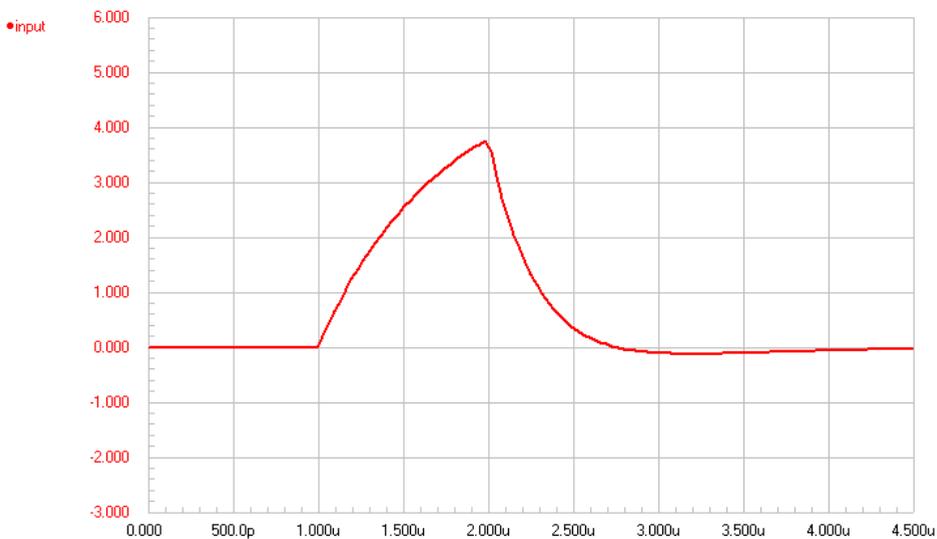


Рис. 4.35. Пример сигнала экспоненциальной формы

Таблица 4.11

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, INPUT)
DC (V или A)	Смещение по постоянному току, используемое при расчете рабочих точек
AC (V или A)	Этот параметр устанавливается в случае, если планируется проводить малосигнальный анализ по переменному току (типовое значение 1 В)
AC Phase (градусы)	Фаза источника напряжения или тока при проведении малосигнального анализа (в градусах)
Initial Value (V или A)	Напряжение или ток в начальный момент времени (например, 0)
Pulse Value (V или A)	Максимальная амплитуда выходного колебания (например, 5)
Rise Delay (с)	Время нарастания от начального напряжения (Initial Value) до напряжения импульса (Pulse Value) (например, 1μ)
Rise Time (с)	Постоянная времени установления (например, 700n)
Fall Delay (с)	Время спада напряжения от значения в импульсе (Pulse Value) до начального напряжения (Initial Value), должно быть больше 0 (например, 2μ)
Fall Time (с)	Постоянная времени восстановления (например, 300n)

Задание сигнала экспоненциальной формы

Ниже приведено описание сигнала, где t — текущий отсчет времени.

$$V(t_0 \text{ to } t_{RD}) = VI$$

$$V(t_{RD} \text{ to } t_{FD}) = VI + (VP - VI) (1 - e^{-(t - t_{RD}) / \tau_{RT}})$$

$$V(t_{FD} \text{ to } t_{STOP}) = VI + (VP - VI) (-e^{-(t - t_{RD}) / \tau_{RT}}) + (VI - VP) (1 - e^{-(t - t_{FD}) / \tau_{FT}})$$

Initial Amplitude (VI)

Начальная амплитуда выходного сигнала (измеренная в В или А), относительно отрицательного вывода, обычно соединенного с землей.

Pulse Amplitude (VP)

Максимальная амплитуда выходного колебания (в В или А).

Rise Time Delay (RD)

Точка времени, относительно нулевого момента, в которой начинается нарастание сигнала. Этот параметр обеспечивает сдвиг фазы путем задержки начала экспоненциального сигнала.

Rise Time Constant (RT)

Стандартное значение постоянной времени установления.

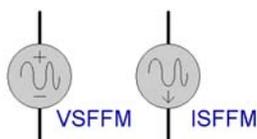
Fall Time Delay (FD)

Точка времени, относительно нулевого момента, в которой начинается спад сигнала.

Fall Time Constant (FT)

Стандартное значение постоянной времени восстановления.

Источник частотно модулированного сигнала



Данный источник используется для получения частотно-модулированного (ЧМ) сигнала (рис. 4.36). После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.12.

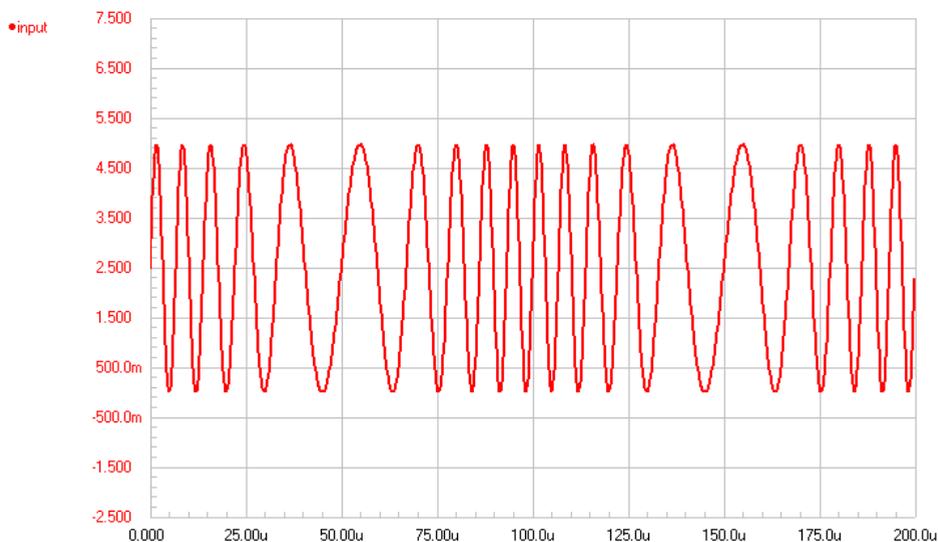


Рис. 4.36. Пример ЧМ сигнала.

Таблица 4.12

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, INPUT)
DC (V или A)	Смещение по постоянному току, используемое при расчете рабочих точек
AC (V или A)	Этот параметр устанавливается в случае, если планируется проводить малосигнальный анализ по переменному току (типовое значение 1 В)
AC Phase (градусы)	Фаза источника напряжения или тока при проведении малосигнального анализа
Offset (V или A)	Постоянное смещение генератора сигнала (например, 2.5)
Amplitude (V или A)	Пиковая амплитуда источника напряжения или тока (например, 2.5)
Carrier (F)	Частота несущей (например, 100k)
Modulation	Индекс модуляции (например, 5)
Signal (F)	Индекс модуляции (например, 5)

Описание ЧМ сигнала

Ниже приведено описание сигнала, где t — текущий отсчет времени.

$$V(t) = V_0 + V_A \sin(2\pi F_c t + M D I \sin(2\pi F_s t))$$

DC Offset (V_0)

Начальное смещение выходного сигнала (измеренная в В или А), относительно отрицательного вывода, обычно соединенного с землей.

Peak Amplitude (V_A)

Максимальная амплитуда выходного колебания без учета постоянной составляющей, измеряется в В или А.

Carrier Frequency (F_c)

Частота немодулированного сигнала в Гц.

Modulation Index (MDI)

Индекс модуляции, определяемый как функция амплитуды модулирующего сигнала и показывающий уровень модуляции.

$$MDI = (\text{девиация частоты}) / F_s$$

Signal Frequency (F_s)

Частота модулирующего сигнала в Гц.

Линейные зависимые источники

Программа моделирования поддерживает стандартные SPICE модели линейных зависимых источников. Любой из этих источников имеет два входа и два выхода. Напряжение или ток на выходах описывается является линейной функцией от напряжения или тока на входах, и определяется коэффициентом усиления, проводимостью или сопротивлением источника.

Размещение линейных зависимых источников на схеме производится из библиотеки символов моделирования, расположенной в библиотечной базе данных **\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**. Рядом с изображением приводится библиотечное имя источника.

Источник тока, управляемый напряжением

После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.13.



Таблица 4.13

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, GSRC1)
Part Type	Проводимость источника в См

Источник напряжения, управляемый напряжением

После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.14.

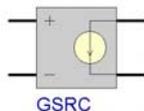


Таблица 4.14

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, ESRC1)
Part Type	Усиление по напряжению источника

Источник тока, управляемый током

После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.15.

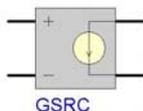


Таблица 4.15

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, FSRC1)
Part Type	Усиление по току источника

Источник напряжения, управляемый током

После размещения этого источника на схеме необходимо задать следующие атрибуты, представленные в таблице 4.16.

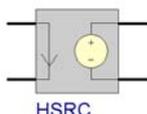


Таблица 4.16

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, HSRC1)
Part Type	Сопротивление источника в Ом

Преобразователь частота/напряжение

После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.17.

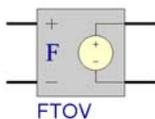


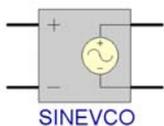
Таблица 4.17

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, FTOV1)
VIL	Нижний уровень входного порога
VIH	Верхний уровень входного порога
CYCLES	Количество циклов на В выходного сигнала

Управляемый напряжением генератор синусоидального сигнала

После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.18.

Таблица 4.18



Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, SINVC01)
LOW	Нижний уровень входного порога
HIGH	Верхний уровень входного порога
C1	Точка входного управляющего напряжения 1
C2	Точка входного управляющего напряжения 2
C3	Точка входного управляющего напряжения 3
C4	Точка входного управляющего напряжения 4
C5	Точка входного управляющего напряжения 5
F1	Точка частоты сигнала на выходе 1
F2	Точка частоты сигнала на выходе 2
F3	Точка частоты сигнала на выходе 3
F4	Точка частоты сигнала на выходе 4
F5	Точка частоты сигнала на выходе 5

Управляемый напряжением генератор прямоугольных импульсов

После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.19.

Таблица 4.19



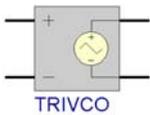
Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, SQRVCO1)
LOW	Пиковое значение низкого уровня выходного сигнала (по умолчанию 0)
HIGH	Пиковое значение высокого уровня выходного сигнала (по умолчанию 5)
CYCLE	Коэффициент заполнения [0,1] (по умолчанию 0.5)
RISE	Время нарастания [0,] (по умолчанию 1u)
FALL	Время спада [0,] (по умолчанию 1u)

C1	Точка входного управляющего напряжения 1 (по умолчанию 0)
C2	Точка входного управляющего напряжения 2 (по умолчанию 1)
C3	Точка входного управляющего напряжения 3 (по умолчанию 2)
C4	In Точка входного управляющего напряжения 4 (по умолчанию 3)
C5	Точка входного управляющего напряжения 5 (по умолчанию 4)
F1	Точка частоты сигнала на выходе 1 [0,] (по умолчанию 0)
F2	Точка частоты сигнала на выходе 2 [0,] (по умолчанию 1к)
F3	Точка частоты сигнала на выходе 3 [0,] (по умолчанию 2к)
F4	Точка частоты сигнала на выходе 4 [0,] (по умолчанию 3к)
F5	Точка частоты сигнала на выходе 5 [0,] (по умолчанию 4к)

Управляемый напряжением генератор треугольных импульсов

После размещения этого источника необходимо задать атрибуты, представленные в таблице 4.20.

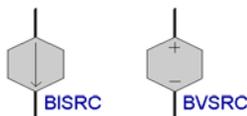
Таблица 4.20



Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, TRIVCO1)
LOW	Пиковое значение низкого уровня выходного сигнала (по умолчанию -1)
HIGH	Пиковое значение высокого уровня выходного сигнала (по умолчанию 1)
CYCLE	Коэффициент заполнения [0,1] (по умолчанию 0.5)
C1	Точка входного управляющего напряжения 1 (по умолчанию 0)
C2	Точка входного управляющего напряжения 2 (по умолчанию 1)

C3	Точка входного управляющего напряжения 3 (по умолчанию 2)
C4	Точка входного управляющего напряжения 4 (по умолчанию 3)
C5	Точка входного управляющего напряжения 5 (по умолчанию 4)
F1	Точка частоты сигнала на выходе 1 [0,] (по умолчанию 0)
F2	Точка частоты сигнала на выходе 2 [0,] (по умолчанию 1к)
F3	Точка частоты сигнала на выходе 3 [0,] (по умолчанию 2к)
F4	Точка частоты сигнала на выходе 4[0,] (по умолчанию 3к)
F5	Точка частоты сигнала на выходе 5 [0,] (по умолчанию 4к)

Нелинейные зависимые источники



Программа моделирования поддерживает стандартные SPICE модели нелинейных зависимых источников напряжения и тока. Модели источников находятся в библиотеке, расположенной по адресу **Program Files\Design Explorer 99 SE\Library\Sch\Sim\Simulation Symbols.lib**. После размещения этого источника на схеме необходимо задать атрибуты, представленные в таблице 4.21.

Таблица 4.21

Атрибут	Значение
Designator	Задается требуемое имя источника сигнала (например, GSRC1)
Part Type	Выражение определяющее форму сигнала источника (например, V(IN)^2)

Задание формы сигнала

Напряжение или ток сигнала описывается следующим образом:

V= выражение or I= выражение

Значение V или I определяет напряжение на выводах устройства или ток через него. Выражение, используемое для V или I может быть функцией напряжения или тока, протекающих через источники напряжений в системе.

Для того чтобы в выражении использовать ссылку на узел схемы, необходимо сначала определить имя узла при помощи объекта **Net Label**. После этого имя, определенное в поле **Net** атрибутов объекта **Net Label**, можно использовать в качестве ссылки на узел при помощи следующего синтаксиса:

$V(net)$ ссылка на напряжение в узле **Net**

$I(net)$ ссылка на ток в узле **Net**

Примеры

Например, если имеется узел схемы, помеченный при помощи объекта **Net Label** именем **IN**, то в поле **Part Type** источника могут быть введены следующие верные значения:

$V(IN)^3$

$COS(V(IN))$

По умолчанию, узел имеет ссылку на имя **Spice Reference Net Name**, которое задается в диалоговом окне **Analog Option** (по умолчанию **GND**). Существует возможность задания различных опорных узлов напрямую в выражении, используя следующий синтаксис:

$V(netlabel1,netlabel2)$

Например,

$LN(COS(LOG(V(NetLabel1,NetLabel2)^2)))-V(NetLabel2)^V(NetLabel1)$

Поддерживаемые функции

ABS	LN	LOG	EXP
SIN	ASIN	ASINH	SINH
COS	ACOS	ACOSH	COSH
TAN	ATAN	ATANH	SQRT

Стандартные операторы

+ — * / ^

Примечание

Если аргумент функций **LOG**, **LN** или **SQRT** становится меньше нуля, то используется его абсолютное значение. Если делитель становится равным нулю или аргумент **LOG** или **LN** становится нулевым, то генерируется сообщение об ошибке. Ряд проблем может также возникнуть в случаях, когда аргумент функции входит в область, где функция не определена, и расчет ее частной производной невозможен.

Компоненты и модели

Для описания аналоговых устройств, расположенных на принципиальных схемах, программа моделирования использует стандартный синтаксис программы SPICE. В большинстве случаев нужно просто разместить тот или иной компонент из библиотеки, установить его номинал (например, значение сопротивления для резистора) и запустить процесс моделирование. Любой компонент содержит всю необходимую для моделирования информацию, включая префиксы позиционных обозначений и нумерацию выводов в многосекционных компонентах.

Программа SPICE также поддерживает множество расширенных функций, позволяющих более точно моделировать поведение компонентов. Например, может возникнуть необходимость промоделировать напыленный резистор, основываясь на геометрической информации о нем или определить температурный диапазон работы транзистора.

Эта дополнительная информация вводится в поля **Part Fields** компонента после его размещения на листе принципиальной схемы. Описание того, какая информация вводится в эти поля для разных типов устройств, приведено в подразделе *Описания устройств*.

Существует возможность создавать свои собственные символы элементов схем для моделирования. Связь символов элементов с моделями описана в разделе *Создание собственных компонентов и подготовка их к моделированию* данной главы.

Библиотеки символов компонентов

Программа моделирования включает широкий набор библиотек символов элементов для моделирования схем. Эти библиотеки охватывают большое количество устройств, наиболее часто используемых при проектировании схем. Символы элементов группируются в библиотеках по функциональному назначению. В таблице 4.22 представлен полный перечень таких библиотек.

Таблица 4.22

74XX.LIB	JFET.lib	Simulation Symbols.lib
7 Segment Displays.lib	Math.lib	Timer.lib
BJT.lib	Mesfet.lib	Transformer.lib
Buffer.lib	Mosfet.lib	Transmission lines.lib
CAmp.lib	OpAmp.lib	Tube.lib
CMOS.lib	Opto.lib	UJT.lib
Comparator.lib	Regulator.lib	Crystal.lib
Diode.lib	Relay.lib	Misc.lib
Fuse.lib	SCR.lib	Triac.lib
IGBT.lib	Switch.lib	

Библиотеки символов находятся в библиотечной базе данных по адресу **Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**. Каждый компонент в этих библиотеках содержит ссылку на модель или подсхему.

Поиск компонентов в библиотеках

Функция поиска является мощным инструментом для поиска нужного компонента в библиотеках символов элементов. Поиск можно производить по библиотечным ссылкам (**Library Reference**) (имя компонента в данной библиотеке) или по описанию компонентов (**Description**), а также по обоим этим параметрам.

Для инициации процесса поиска компонента нужно вызвать диалоговое окно **Find Schematic Component** с помощью команды меню **Tools » Find Component**.

Советы по поиску компонентов

Поиск при помощи поля **Library Reference**, как правило, работает быстрее.

Так как различные производители используют разные префиксы и суффиксы при именовании своих изделий, для расширенного поиска рекомендуется использовать символ-заменитель "*".

Для увеличения шансов совпадения полезно использовать поле **Description**, причем, здесь лучше задавать сложные условия поиска из нескольких слов, окруженных символами-заменителями "*" и разделенных пробелами (например, ***barrel* *shifter***).

Если проведенный поиск не дал результатов, проверьте правильно ли задан путь поиска в поле **Path**. Для того чтобы убедиться, что все сделано правильно, попробуйте найти компонент, заведомо существующий в какой-либо библиотеке.

Для подключения библиотеки с найденным компонентом к системе необходимо использовать кнопку **Add to Library List**.

Модели и подсхемы

Механизм моделирования программы SPICE имеет встроенные модели для следующих типов аналоговых компонентов: резисторов, конденсаторов, катушек индуктивности, катушек трансформаторов с индуктивной связью, независимых и управляемых источников напряжения и тока, линий передачи с потерями и без таковых, переключателей, равномерно распределенных RC линий, а также для пяти наиболее часто встречающихся полупроводниковых устройств: диодов, биполярных транзисторов, полевых канальных транзисторов, полевых транзисторов с барьером Шоттки и МОП-транзисторов. Программа моделирования включает также большое количество файлов моделей, которые определяют поведение специфических экземпляров перечисленных устройств. Эти файлы моделей обычно имеют расширение **MDL**.

Программа моделирования также поддерживает описания более сложных устройств, таких как операционные усилители, стабилизаторы, генераторы синхроимпульсов, кварцевые генераторы и т. д., используя иерархический синтаксис описания подсхем. Подсхема состоит из элементов программы SPICE, которые описываются аналогично моделям простых элементов. Не существует никаких ограничений на размер и сложность подсхем, и одна подсхема может содержать другую. Каждая подсхема определяется в специальном файле с расширением **CKT**, который также часто обозначается термином "модель".

Описания устройств

Для дискретных компонентов (таких как резисторы), для ссылки на конкретную модель элемента используется поле **Part Type**. Для недискретных компонентов (операционные усилители и т.д.) содержимое этого поля нужно изменить и вписать сюда ссылку на какую-либо другую модель.

Резисторы (Resistors)

Кроме стандартных резисторов программа моделирования поддерживает также полупроводниковые резисторы. Они позволяют оценить сопротивление как функцию геометрического представления резистора и задать температуру, в которой будет функционировать устройство. Символы резисторов находятся в библиотеке символов элементов для моделирования, которая располагается в библиотечной базе данных по адресу **Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**.

Таблица 4.23

Атрибут	Значение
Designator	Позиционное обозначение резистора
Part Type	Номинал сопротивления — если это значение установлено, то оно имеет больший приоритет, чем геометрическое представление
L #	Длина резистора (в метрах)
W #	Ширина резистора (в метрах)
Temp #	Температура, в которой это устройство функционирует (в °C, по умолчанию равно 27°C)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Конденсатор (Capacitor)

Кроме стандартных конденсаторов программа моделирования поддерживает также полупроводниковые конденсаторы. Они позволяют промоделировать емкость как функцию геометрического представления конденсатора и установить начальное значение емкости. Символы конденсаторов находятся в библиотеке символов элементов для моделирования, которая располагается в библиотечной базе данных по адресу `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`.

Таблица 4.24

Атрибут	Значение
Designator	Позиционное обозначение конденсатора
Part Type	Номинал конденсатора — если это значение установлено, то оно перекрывает геометрическое определение
L #	Длина конденсатора (в метрах)
W #	Ширина конденсатора (в метрах)
IC	Начальные условия — напряжение на конденсаторе в нулевой момент времени. Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Катушка индуктивности (Inductor)

Символы катушки индуктивности находятся в библиотеке символов элементов для моделирования, которая располагается в библиотечной базе данных по адресу **\\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**. Атрибуты индуктивности представлены в таблице 4.25.

Таблица 4.25

Атрибут	Значение
Designator	Позиционное обозначение индуктивности
Part Type	Номинал индуктивности
IC	Начальные условия — значение тока, протекающего через индуктивность в нулевой момент времени. Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.

Связанные катушки индуктивности (Coupled Inductors)

Катушки индуктивности могут быть связаны с помощью стандартного синтаксиса описания связанных устройств для программы SPICE. Этот процесс описан в подразделе *Включение дополнительной SPICE информации в список соединений* раздела *Подготовка и проведение моделирования*.

Диод (Diode)

Символы диода и стабилитрона находятся в библиотеке **\\Program Files\Design Explorer 99 SE\Library\Sim\Diode.ddb**. Атрибуты стандартного диода представлены в таблице 4.26.

Таблица 4.26

Атрибут	Значение
Designator	Позиционное обозначение диода
Part Type	Тип диода
Area #	Число параллельно соединенных моделей устройств данного типа
Off / On #	Off означает установку в ноль падение напряжения на выводах диода во время расчета рабочей точки, помогает при отсутствии сходимости анализа.
IC	Начальные условия — напряжение на диоде в нулевой момент времени. Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.
Temp (°C) #	Температура, в которой это устройство функционирует (по умолчанию равно 27°C)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Транзисторы

Биполярный (BJT)

Модель для такого транзистора основана на модели Гаммеля-Пуна (Gummel-Poon) с интегральным законом заряда. Однако, если ряд параметров для модели Гаммеля-Пуна не заданы, то она упрощается до модели Эберса-Молла (Ebers-Moll). Символы биполярных транзисторов находятся в библиотеке `\Program Files\Design Explorer 99 SE\Library\Sch\Sim\Bjt.lib`. Атрибуты транзистора представлены в таблице 4.27.

Таблица 4.27

Атрибут	Значение
Designator	Обозначение транзистора
Part Type	Тип транзистора
Area #	Число параллельно соединенных моделей устройств данного типа
Off / On #	Off означает установку в ноль напряжения на выводах во время анализа в рабочей точке, помогает при отсутствии сходимости анализа.
IC	Начальные условия — напряжения VBE и VCE в нулевой момент времени (два значения, разделенные запятой). Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.
Temp (°C) #	Температура, в которой это устройство функционирует (по умолчанию равно 27°C)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Полевой транзистор с управляющим p-n переходом (JFET)

Эта модель основана на модели канального транзистора Шихмана-Ходжеса (Shichman-Hodges). Символы транзистора этого типа находятся в библиотеке **Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**. Атрибуты транзистора представлены в таблице 4.28.

Таблица 4.28

Атрибут	Значение
esignator	Обозначение транзистора
Part Type	Тип транзистора
Area #	Число параллельно соединенных моделей устройств данного типа
Off / On #	Off означает установку в ноль напряжения на выводах во время анализа в рабочей точке, помогает при отсутствии сходимости анализа.
IC	Начальные условия — напряжения VDS и VGS в нулевой момент времени (два значения, разделенные запятой). Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.
Temp (°C) #	Температура, в которой это устройство функционирует (по умолчанию равно 27°C)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

МОП-транзистор (MOSFET)

Программа моделирования поддерживает следующие модели транзисторов с изолированным затвором (MOSFET): Шихмана-Ходжеса; BSIM 1, 2, 3 и MOS 2, 3 и 6. Символы таких транзисторов находятся в библиотеке `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`. Атрибуты транзисторов представлены в таблице 4.29.

Таблица 4.29

Атрибут	Значение
Designator	Обозначение транзистора
Part Type	Тип транзистора
L #	Длина канала (в метрах)
W #	Ширина канала (в метрах)
AD #	Площадь стока (в метрах в квадрате)
AS #	Площадь истока (в метрах в квадрате)
PD #	Периметр стокового перехода (в метрах)
PS #	Периметр истокового перехода (в метрах)
NRD #	Коэффициент эмиссии стока
NRS #	Коэффициент эмиссии истока
OFF #	Off означает установку в ноль напряжения на выводах во время расчета рабочей точки, помогает при отсутствии сходимости анализа.
IC	Начальные условия — напряжения VDS, VGS и VBS в нулевой момент времени (три значения, разделенные запятой). Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.
Temp (°C) #	Температура, в которой это устройство функционирует (по умолчанию равно 27°C)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Полевой транзистор с барьером Шоттки (MESFET)

Модель этого типа транзисторов происходит от модели Стаца (Statz) полевых транзисторов на основе арсенида галлия (GaAsFET). Символы таких транзисторов находятся в библиотеке **\\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**. Атрибуты транзисторов представлены в таблице 4.30.

Таблица 4.30

Атрибут	Значение
Designator	Обозначение транзистора
Part Type	Тип транзистора
Area #	Число параллельно соединенных моделей устройств данного типа
Off / On #	Off означает установку в ноль напряжения на выводах во время расчета рабочей точки, помогает при отсутствии сходимости анализа.
IC	Начальные условия — напряжения VDS и VGS в нулевой момент времени (два значения, разделенные запятой). Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Переключатели, управляемые напряжением или током (Voltage/Current controlled switch)

Программа моделирования имеет некоторое количество встроенных переключателей с различными пороговыми напряжениями и сопротивлениями в различных режимах. Символы для этих переключателей находятся в библиотеке **\\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**. Каждый компонент связан с моделью, имеющей такое же имя, которая расположена в папке **\\Program Files\Design Explorer 99 SE\Library\Sim\Switches**.

Если необходимо определить собственную модель переключателя, то в качестве ссылок можно использовать модели из этой папки. Таблица 4.31 описывает все переключатели, содержащиеся в библиотеке.

Таблица 4.31

Библиотечный компонент	Описание
CSW	Переключатель, управляемый током (используется по умолчанию)
SW	Переключатель, управляемый напряжением (используется по умолчанию)
SW05	$V_T=500.0m$
SWM10	$V_T=-0.01$
SWP10	$V_T=0.01$
STTL	$V_T=2.5 V_H=0.1$
TTL	$V_T=2 V_H=1.2 ROFF=100E+6$
TRIAC	$V_T=0.99 RON=0.1 ROFF=1E+7$

Ядро анализа программы SPICE поддерживает следующие параметры переключателей:

Переключатель, управляемый напряжением (Voltage controlled switch)

Таблица 4.32

Название	Параметры и единицы измерения	Значение по умолчанию
V_T	Пороговое напряжение (В)	0.0 #
V_H	Напряжение петли гистерезиса (В)	0.0
RON	Сопротивление замкнутого ключа (Ом)	1.0
$ROFF$	Сопротивление разомкнутого ключа (Ом)	$1/GMIN^*$

Переключатель, управляемый током (Current controlled switch)

Таблица 4.33

Название	Параметры и единицы измерения	Значение по умолчанию
I_T	Пороговый ток (А)	0.0
I_H	Ток петли гистерезиса (А)	0.0
RON	Сопротивление замкнутого ключа (Ом)	1.0
$ROFF$	Сопротивление разомкнутого ключа (Ом)	$1/GMIN^*$

Для приведения в действие переключателя управляющий сигнал должен превысить это напряжение.

* Значение $GMIN$ устанавливается в диалоговом окне Analog Options.

Замечания по использованию переключателей

Использование идеальных элементов с сильно выраженной нелинейностью, таких как переключатели, может привести к возникновению больших скачков напряжения в узлах электрических цепей. Такие скачки, связанные с изменением состояния переключателя, могут стать причиной числовых округлений или отсутствия сходимости, что приводит к трудностям в процессе моделирования при вычислении значений на следующем временном шаге или ошибочным результатам. Следующие действия помогут по возможности предотвратить эти последствия:

- Первым делом установите импеданс переключателя достаточно высоким или низким, чтобы он слабо влиял на другие элементы цепи. Установка значения импеданса близкого к "идеальному" ухудшит проблему скачков напряжения.
- При моделировании реального устройства (например, транзистора) его сопротивление в открытом состоянии должно иметь реальное значение в зависимости от типа моделируемого устройства.
- Если переключатель должен иметь большое отношение сопротивлений в закрытом и открытом состояниях ($ROFF/RON > 1e+12$), тогда приемлемый допуск на ошибки при анализе переходных процессов должен быть уменьшен посредством установки параметра TRTOL в диалоговом окне Analog Options в значение меньшее, чем значение по умолчанию (7.0). Попробуйте установить TRTOL в 1.
- Если переключатель расположен рядом с конденсатором, то значение CHGTOL также должно быть уменьшено, например, до значения 1E-16.

Перечисленные изменения заставляют ядро программы моделирования SPICE обрабатывать схему вблизи переключателей более корректно, что позволяет избежать ошибок, связанных с быстрыми изменениями сигналов в цепях.

Предохранители (Fuses)

Символы для основных типов предохранителей находятся в библиотеке \Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb. Номинальное значение тока для предохранителя определяется в соответствующем поле элемента (**Current Part Field**). Атрибуты предохранителей перечислены в таблице 4.34.

Таблица 4.34

Атрибут	Значение
Designator	Обозначение предохранителя
Current #	Номинальное значение тока срабатывания, например 250 мА
Resistance #	Сопротивление предохранителя (по умолчанию для всех моделей предохранителей равно 0.001 Ом)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Кварцевые резонаторы (Crystals)

Символы кварцевых резонаторов находятся в библиотеке **CRYSTALS**, которая располагается в базе данных `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`. Существует некоторое количество встроенных типов кварцевых резонаторов, а также устройств на их основе. Все типы кварцевых устройств содержат поле резонансной частоты (**Frequency Part Field**), однако нет необходимости вводить значение в это поле, за исключением случаев, когда нужно, чтобы это значение отличалось от значения в поле элемента (**Part Type**). Если используется базовый компонент **XTAL**, который имеет значение частоты по умолчанию равное 1 МГц, тогда нужно ввести требуемое значение в поле частоты элемента (**Frequency Part Field**). Атрибуты кварцевых устройств представлены в таблице 4.35.

Таблица 4.35

Атрибут	Значение
Designator	Обозначение резонатора
Freq #	Основная частота резонатора
RS #	Последовательное сопротивление резонатора
C #	Емкость резонатора
Q #	Значение добротности эквивалентной схемы

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Реле (Relays)

Символы реле находятся в библиотеке **RELAY**, которая располагается в базе данных `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`. Атрибуты реле описаны в таблице 4.36.

Таблица 4.36

Атрибут	Значение
Designator	Обозначение реле
Pullin #	Напряжение срабатывания
Dropoff #	Падение напряжение на контакте
Contact #	Сопротивление контактов (Ом)
Resistance #	Сопротивление катушки (Ом)
Inductance #	Индуктивность катушки (Гн)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Трансформаторы

Символы трансформаторов находятся в библиотеке **TRANSFORMER**, которая располагается в базе данных `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`. Эти символы связаны с двумя типами моделей — модель трансформаторов, которая учитывает намагничивание и индуктивность рассеяния (**TRANS** и **TRANSCT**), и модель связанных катушек индуктивности, которая рассматривает трансформатор как пару связанных катушек индуктивности (**KTRANSFORMER** и **KTRANSFORMERCT**).

Атрибуты трансформаторов **TRANS** и **TRANSCT** описаны в таблице 4.37.

Таблица 4.37

Атрибут	Значение
Designator	Обозначение трансформатора
Ratio #	Коэффициент трансформации — отношение числа витков вторичной обмотки к числу первичной обмотки
RP #	Сопротивление первичной обмотки (по умолчанию 0.1 Ом)
RS #	Сопротивление вторичной обмотки (по умолчанию 0.1 Ом)
Leak #	Индуктивность рассеяния (по умолчанию 1 мкГн)
Mag #	Индуктивность намагничивания (по умолчанию 1мкГн)

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Линии передачи (Transmission Lines)

Программа моделирования поддерживает линии передачи без потерь, с распределенными потерями и линии передачи с сосредоточенными RC потерями. Символы линий передачи находятся в библиотеке **TRANSLINE**, которая располагается в базе данных `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb`. Атрибуты различных типов линий передачи включают:

Линии передачи без потерь

Таблица 4.38

Атрибут	Значение
Designator	Обозначение линии передачи
ZO #	Волновое сопротивление (Ом)
TD #	Задержка передачи (см. замечания ниже)
F	Частота (см. замечания ниже)
NL	Относительная электрическая длина линии передачи на частоте F (см. замечания ниже)
IC	Начальные условия — напряжение и ток для каждого порта линии передачи. Действует только при установленной опции Use Initial Conditions в диалоговом окне Setup Analyses.

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Определение длины линии передачи без потерь

Длина такой линии должна быть выражена двумя способами:

Непосредственное задание времени задержки передачи (например, $TD = 10$ нс).

Задается частота F и относительная электрическая длина NL . Если F определена, а NL нет, то значение NL устанавливается по умолчанию равным **0.25** (то есть равным четверти длины волны).

Линия передачи с потерями

Используется двухпортовая сверточная модель для однопроводной линии передачи с потерями. Заметим, что описываемая модель с установленными в ноль параметрами потерь может быть более точной, чем модель линии без потерь, так как здесь учитывается большее число параметров.

Модель линии с потерями содержит атрибуты для задания сопротивления, индуктивности, емкости и длины. Все эти параметры передать непосредственно из элемента принципиальной схемы невозможно, однако можно создать собственный файл модели для ссылки на него. Для этого нужно скопировать файл **LTRA.MDL**, ввести новое имя для него и изменить строку, расположенную сразу после строки **.MODEL**, так, чтобы она совпадала с именем нового файла, затем отредактировать нужные атрибуты. Для того, чтобы использовать новую модель в вашем проекте, необходимо ввести ее имя в поле **Part Type** компонента.

Например, из существующего файла **LTRA.MDL**:

```
.MODEL LTRA LTRA(R=0.000 L=9.130n C=3.650p LEN=1.000)
```

Можно создать новый файл **ltra10.MDL**:

```
.MODEL LTRA10 LTRA(R=0.2 L=32n C=13p LEN=10.000)
```

RC линии передачи с равномерно распределенными потерями

Таблица 4.39

Атрибут	Значение
Designator	Обозначение линии передачи
L #	Длина RC линии (метры)
N #	Количество сосредоточенных сегментов, используемых при моделировании RC линии

Эти опции имеют значения по умолчанию и устанавливаются только по необходимости.

Цифровые компоненты

Программа моделирования поддерживает специальный описательный язык, который позволяет моделировать цифровые устройства, используя расширенную, управляемую событиями версию программы XSPICE. Цифровые устройства моделируются с помощью языка Digital SimCode™. Более подробно этот язык описан в разделе *Моделирование цифровых схем*.

В библиотечной базе данных `\Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb` размещены две библиотеки цифровых компонентов — `74xx.lib` и `CMOS.lib`.

Модели для компонентов в этих библиотеках содержат значения по умолчанию для всех атрибутов — задержка распространения, нагрузка и т.д. Если требуется изменить эти значения для какого-либо отдельного компонента, тогда нужно заполнить соответствующие поля **Part Fields** в диалоговом окне **Component's Part**.

Таблица 4.40

Атрибут	Значение
Propagation #	Задержка распространения. Задается значение min или max вместо используемого по умолчанию.
Loading #	Входная нагрузочная характеристика. Задается значение min или max вместо используемого по умолчанию.
Drive #	Выходная нагрузочная характеристика. Задается значение min или max вместо используемого по умолчанию.
Current #	Ток потребления устройства, определяющее потребляемую мощность. Задается значение min или max вместо используемого по умолчанию.
PWR value #	Напряжение питания, подаваемое на вывод питания устройства.
GND value #	Напряжение заземления. Это напряжение должно быть определено, если задано значение в поле PWR Part Field.
VIL value #	Входное напряжение низкого уровня.
VIH value #	Входное напряжение высокого уровня.
VOL value #	Выходное напряжение низкого уровня.
VOH value #	Выходное напряжение высокого уровня.
WARN	Устанавливается в ON для сигнализации ошибок: во времени вхождения в режим, во времени занятости устройства, во времени восстановления, в ширине импульса, выход за пределы допустимых частот, выход за пределы допустимого напряжения питания. Предупреждения записываются в файл с расширением SIM, ошибки — в файл с расширением ERR.

Создание собственных компонентов и подготовка их к моделированию

Первым шагом в создании собственных компонентов для моделирования является рисование их символов в редакторе библиотек элементов принципиальных схем. После создания символа можно ввести информацию по моделированию в библиотечные поля (**Library Fields**) данного компонента (команда **Tools** » **Description** из меню библиотечного редактора). Эта информация предоставляет сведения программе моделирования о типе компонента, о месте нахождения соответствующей ему модели, о назначении и номерах выводов различных секций составных компонентов, а также специфическую SPICE информацию.

Поля **Library Fields** и **Part Fields**

Программа моделирования считывает SPICE информацию и данные списка соединений из библиотечных текстовых полей (**Library Fields**), которые еще называются неизменяемыми полями (**Read-Only Fields**), так как изменение их содержимого не доступно в редакторе принципиальных схем, возможно только в редакторе библиотек, и из текстовых полей (**Part Fields**). Не все компоненты требуют определять одинаковые данные в текстовых полях, например, дискретные компоненты не имеют ссылки на модель. Обратившись к библиотекам моделей (база данных **Program Files\Design Explorer 99 SE\Library\Sim\Sim.ddb**) можно узнать, какую информацию необходимо задавать для каждого конкретного типа компонента. Ниже показано, какую информацию нужно вносить в библиотечные текстовые поля (**Library Fields**).

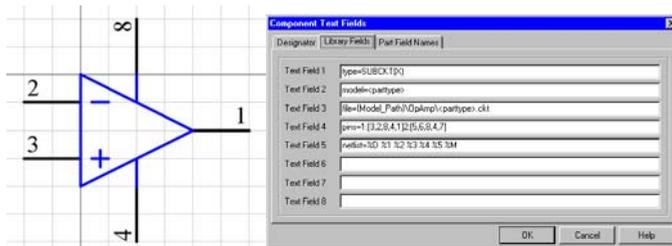


Рис. 4.37. Информация о модели заносится в поля **Library Fields** в редакторе библиотек.

Library Field 1

type=<Device Type>(<SPICE Prefix>)

Здесь задается тип моделируемого устройства (**Device Type**) и SPICE префикс (**SPICE Prefix**), например,

type=NPN(Q)

NPN показывает, что данное устройство является биполярным p-p-n транзистором. Параметр **Q** является префиксом, необходимым для программы SPICE (таблица 4.41).

Таблица 4.41

Тип устройства и SPICE префикс	Функция
CAP(C)	Конденсатор
CCCS(F)	Источник тока, управляемый током
CCS(W)	Переключатель, управляемый током
CCVS(H)	Источник напряжения, управляемый током
DIODE(D)	Диод
INDUCTOR(L)	Катушка индуктивности
IPULSE(I)	Импульсный источник тока
IPWL(I)	Источник тока кусочно-линейной формы
ISFFM(I)	Источник тока с частотной модуляцией одной частотой
ISIN(I)	Источник тока синусоидальной формы
ISRC(I)	Источник постоянного тока
LTRA(O)	Линия передачи с потерями
MUTUALINDUCTANCE(K)	Катушки индуктивности с индуктивной связью
NMOS(M)	МОП транзистор с обедненным n-каналом
NEMOS(M)	МОП транзистор с обогащенным n-каналом
NJFET(J)	Полевой транзистор с n-каналом
NMESFET(Z)	Полевой транзистор с барьером Шоттки с n-каналом
NPN(Q)	Биполярный n-p-n транзистор
PDMOS(M)	МОП транзистор с обедненным p-каналом
PEMOS(M)	МОП транзистор с обогащенным p-каналом
PJFET(J)	Полевой транзистор с p-каналом
PMESFET(Z)	Полевой транзистор с барьером Шоттки с p-каналом
PNP(Q)	Биполярный p-n-p транзистор
POT(R)	Переменный резистор или потенциометр
RES(R)	Резистор
SEMICAP(C)	Полупроводниковый конденсатор
SEMIRES(R)	Полупроводниковый резистор
SIMCODE(A)	Цифровое устройство, описанное SimCode
TRA(T)	Линия передачи без потерь
UDRC(U)	Равномерно распределенная RC линия (с потерями)
VCCS(G)	Источник тока, управляемый напряжением
VCSW(S)	Переключатель, управляемый напряжением
VCVS(E)	Источник напряжения, управляемый напряжением
NLDS(B)	Нелинейный зависимый источник
VPULSE(V)	Импульсный источник напряжения
VPWL(V)	Источник напряжения кусочно-линейной формы
VSFFM(V)	Источник напряжения с частотной модуляцией
VSIN(V)	Источник синусоидального напряжения
VSRC(V)	Источник постоянного напряжения
ZENER(A)	Стабилитрон, описанный SimCode
SUBCKT(X)	Подсхема

Library Field 2

model=<model name>

Здесь определяется имя модели, используемой при моделировании данного устройства. Если здесь введена строка model=<parttype>, то в качестве имени модели используется содержимое поля **Part Type** данного компонента.

Library Field 3

file={model_path}\...\<filename>.*

Показывает местоположение файла, где размещена модель, определенная в предыдущем поле. Если используется синтаксис model=<parttype>, тогда это поле обязательно должно быть file={model_path}\...\<parttype>.* Обычно расширением файла является **CRT** или **MDL**. Содержимое файла модели вставляется в конец файла списка соединений.

Library Field 4

Pins=1:[pin1<,pin2,pin3,...>]<2:[pin1<,pin2,pin3,...>]...>

Определяет выводы для каждой секции элемента данного компонента. Порядок следования выводов здесь не фиксированный, но удобнее вводить номера выводов в порядке, который требуется для моделирования. Таким образом упрощается определение номеров и позиций выводов в следующем поле (**Library Field 5**). Данные о порядке часто выносятся в заголовок файла модели.

Например, для операционного усилителя 741 (4 элемента в корпусе) содержимое этого поля будет выглядеть так:

```
pins=1:[3,2,4,11,1]2:[5,6,4,11,7]3:[10,9,4,11,8]4:[12,13,4,11,14]
```

Library Field 5

netlist=<SPICE Data>|<SPICE Data line 2>|...

Это поле определяет, в каком виде данный компонент будет присутствовать в файле списка соединений программы SPICE. Если нужно определить более одной строки данных списка соединений, необходимо вставлять между ними символ вертикальной черты в качестве разделителя.

Кроме непосредственного ввода SPICE информации в этом поле можно определить ссылку на любое другое поле из 16 библиотечных полей. Символ процента (%) обозначает ссылку на другое поле, буква или цифра, следующая за процентом, определяет это поле. Могут быть использованы следующие опции:

%D – Device designation

Здесь указывается обозначение устройства. Если первый символ обозначения не соответствует SPICE префиксу, тогда этот префикс автоматически вставляется в начало строки в списке соединений.

%1, %2, %3, .. %n

Выводы устройства должны быть добавлены в список соединений в порядке, который требуется для SPICE модели. Эти порядковые номера напрямую не используются, каждый используется в качестве номера вывода, который определен в поле (**Library Field 4**) данного элемента.

Таблица 4.42

Параметр	Действие	Результат в списке соединений
%D	Вставка обозначения и, если требуется, добавление SPICE префикса	XU1C
%1	Поиск 1-го вывода 3-го элемента (U1C) в строке pins= и добавление цепи этого вывода в список соединений	NetOnPin10
%2	Поиск 2-го вывода 3-го элемента (U1C) в строке pins= и добавление цепи этого вывода в список соединений	NetOnPin9
%3	Поиск 3-го вывода 3-го элемента (U1C) в строке pins= и добавление цепи этого вывода в список соединений	NetOnPin4
%4	Поиск 4-го вывода 3-го элемента (U1C) в строке pins= и добавление цепи этого вывода в список соединений	NetOnPin11
%5	Поиск 5-го вывода 3-го элемента (U1C) в строке pins= и добавление цепи этого вывода в список соединений	NetOnPin8
%M	Вставка имени модели, определенной в поле Library Field 2	MC4741

Например, микросхема MC4741, содержащая 4 операционных усилителя, с обозначением U1C (третья секция в элементе U1) имеет следующие значения для полей 4 и 5:

```
pins=1:[3,2,4,11,1]2:[5,6,4,11,7]3:[10,9,4,11,8]4:[12,13,4,11,14]
netlist=%D %1 %2 %3 %4 %5 %M
```

Во время генерации списка соединений строка netlist= интерпретируется следующим образом:

Полученный результат будет записан в файл списка соединений:

```
XU1C NetOnPin10 NetOnPin9 NetOnPin4 NetOnPin11 NetOnPin8 MC4741
```

Для описания цифровых устройств используется аналогичный подход, то есть номера, перечисленные в поле **Library Field 5**, рассматриваются как индексы действительных номеров выводов, определенных в поле **Library Field 4**. Как и для аналоговых устройств порядок перечисления выводов в поле 5 должен соответствовать порядку, определенному в модели, описанной SimCode. Отличие этой модели состоит в том,

что входные и выходные выводы перечисляются отдельно друг от друга. Более подробно это описано в пункте *Пример создания условного графического обозначения и SimCode модели устройства 74LS74*.

%F1 ...%F16 — текстовые поля (Text Fields)

%F1 ... %F16 ссылаются на соответствующие текстовые поля **Part Fields 1—16**. Значения из этих полей добавляются в список соединений.

%IF() — добавление в список соединений при соблюдении условия

Содержимое скобок добавляется в список в том случае, если поля, на которые ссылаются обозначения внутри скобок, содержат данные. Эта процедура не рекурсивна (вложенные %IF не допускаются) и распространяется только на содержимое текстовых полей (**Part Field**), то есть допускается %Fх и %Pх. Например:

%IF(AC %F2 %F3) добавит AC 1 0

%IF(PARAMS: %P6) добавит PARAMS: PartField6Name=PartField6Value

%M — Имя модели

Имя модели, указанной в поле **Library Field 2**, добавляется в список соединений.

%P или %P1 ... %P16 — параметры устройства

Добавляется имя текстового поля, а также содержимое этого поля. %P без цифры означает все поля **Part Fields**, которые содержат какие-либо значения, а также их содержимое.

Формат полученной строки:

<TextFieldName1=TextField1> ... <TextFieldName16=TextField16>

Если поле элемента не содержит ни какого значения, тогда оно будет отсутствовать в строке.

%R — библиотечная ссылка

Вставляется содержимое поля библиотечной ссылки (**Library Reference Field**).

%T — тип элемента

В список соединений добавляется содержимое поля **Part Type**, при этом никаких проверок не выполняется.

%V — значение

В список соединений добавляется содержимое поля **Part Type**, при этом требуется, чтобы его значение было численным. Оно может быть целым числом, числом с плавающей точкой, числом с плавающей точкой с последующим целым показателем степени, а также целым числом или числом с плавающей точкой со стандартным множителем. Полный список множителей приведен в разделе *Введение в моделирование*.

Замечание: при использовании множителя не допускаются пробелы между ним и числом, к которому он относится. Символы, не являющиеся стандартными обозначениями множителей, игнорируются.

Library Field 8 (необязательно)

Пример: defaults=F1:0,F2:15,F3:6,...

Это поле применяется для определения параметров по умолчанию, когда они требуются для полей **Part Fields** какого-либо компонента. Программа моделирования считывает соответствующие обозначения из полей элементов и затем добавляет введенные здесь значения.

Например, линия передачи без потерь (библиотечная ссылка **LLTRA**) содержит следующую информацию в поле **Library Field 8**:

```
defaults=F1:50,F2:10ns
```

При составлении списка соединений для этого компонента (между цепями IN, 0 и OUT, 0) будет создана следующая строка:

```
TLLTR1 IN 0 OUT 0 Z0=50 TD=10ns
```

Текстовые поля Part Fields 1 — 16

Эти поля используются для определения параметров компонента после его размещения на схеме. Для добавления этих полей в список соединений в поле Library Field 5 необходимо использовать ранее описанный синтаксис, то есть %F1-16, %P, %P1-16 или **PARAMS**.

Обработка ошибок, возникающих при моделировании

Если при моделировании схем возникают какие-либо проблемы, прежде всего необходимо определить их источник. Ошибки могут быть либо в самой цепи, либо они могут возникать в процессе моделирования. Ниже приведены рекомендации по решению некоторых проблем, возникающих при анализе схем, опираясь на которые пользователь может исправить проект и повторно запустить моделирование.

Ошибки при генерации списка цепей

После запуска программы моделирования с помощью команды меню **Simulate » Run** проводится проверка схемы и генерируется список цепей. Затем список цепей передается в программу SPICE, которая проводит расчет и выдает его результаты.

При возникновении ошибок во время создания списка цепей появляется окно **Error dialog**. Все возникающие ошибки записываются в файл с расширением **ERR** и именем проекта. Файл открывается автоматически после нажатия кнопки **Yes** в окне сообщения об ошибке. Возможные причины возникновения ошибок при создании списка цепей могут быть следующими:

Упомянутый в файле ошибок элемент неправильно подготовлен к процессу моделирования. Двойным щелчком мыши по компоненту проверьте, содержат ли поля **Read Only Field** правильные ссылки на необходимые модели. Если у пользователя возникают вопросы относительно того, какая информация нужна для моделирования схемы с тем или иным компонентом, полезно посмотреть пример с использованием этого элемента.

Файла модели компонента, указанного в поле **Read Only Field 3**, нет в указанном каталоге. Такое может случиться, если библиотеки модели не были установлены (не подключены) или же каким-то образом удалены из исходного каталога, созданного при установке системы. Модели компонентов должны находиться в каталоге **Program Files\Design Explorer 99 SE\Library\Sim**. Каталог должен содержать 28 папок моделей и подсхем. Если это не так, то возможно при инсталляции не была запущена программа Protel Special Libraries Installer. Запуск программы установки библиотек осуществляется из группы программ Protel 99 SE в меню Пуск системы Windows.

Путь к файлу модели, определенный как **{model_path}**, не соответствует действительному расположению модели. Причина этого — случайное перемещение модели в другой каталог на жестком диске. Отметим, что путь к модели хранится в файле **Advsim99.INI**.

Сходимость программы SPICE

Большинство проблем, возникающих при моделировании, связано со сходимостью. Что же в действительности подразумевается под этим термином? Подобно большин-

ству программ моделирования модуль SPICE3 системы Protel, выполняющий расчет режима по постоянному току, использует итерационный процесс решения описывающей схему системы уравнений. Если модулю не удастся рассчитать напряжения и токи в статическом режиме, то дальнейший анализ цепи невозможен.

Для определения постоянных напряжений и токов в рабочей точке цепи на каждом шаге моделирования программа SPICE3 решает систему линейных уравнений, выраженную в матричной форме. Цепь замещается массивом проводимостей, который переписывается в матрицу для формирования уравнения вида $G * V = I$. При наличии в цепи нелинейных элементов программа SPICE для расчета нелинейностей использует итерационный процесс решения системы линейных уравнений. Программа SPICE берет начальное приближение напряжения в узлах схемы, затем на основе проводимостей цепи рассчитывает токи ветвей. После расчета токов ветвей программа SPICE пересчитывает узловые напряжения, и процесс повторяется снова. Этот циклический процесс будет продолжаться до тех пор, пока все узловые напряжения и токи ветвей не достигнут заданной точности вычисления, то есть пока процесс не сойдется.

Однако, если процесс расчета напряжений и токов не сходится за заданное количество итераций, программа SPICE генерирует сообщение об ошибке, например, "**singular matrix**" (вырожденная матрица), "**Gmin stepping failed**" (сбой метода пошагового изменения минимальной проводимости), "**source stepping failed**" (сбой метода пошагового изменения сигналов источников), "**iteration limit reached**" (достигнуто предельное число итераций) и прерывает процесс моделирования. Программа SPICE использует результаты моделирования на данном шаге в качестве начальных приближений для следующего шага. Если процесс не может сойтись при анализе переходных процессов (пошаговое изменение времени), шаг приращения времени автоматически уменьшается и цикл вычислений повторяется. Когда шаг по времени достигнет своего минимального значения модуль SPICE выдаст сообщение "**Timestep too small**" (слишком маленький шаг по времени).

◆ Точность вычисления и максимально допустимое число итераций задается в диалоговом окне **Analog Options**.

Стратегия решения проблем сходимости

Приведенные ниже советы помогут пользователю решить проблемы сходимости. Сначала следует выключить все виды анализа, за исключением расчета рабочих точек. Начать следует с шага 1, а затем обдумать рекомендации по устранению ошибок.

1. Просмотрите одноименный файл ошибок с расширением **ERR**. В нем описаны все встретившиеся ошибки при моделировании.
2. Проверьте правильность прорисовки схемы и связанность. А именно:
 - Убедитесь в корректности соединения проводников схемы. Не допустимы несоединенные узлы и неподключенные элементы.

- Каждая схема должна иметь заземляющий узел, а каждый узел схемы должен иметь путь прохождения тока к земле. Некоторые компоненты, например, трансформаторы и конденсаторы могут изолировать узел от земли. Источники напряжения могут быть подключены к короткозамкнутым цепям, а цепи с источниками тока иметь обрыв.
 - Не путайте математический ноль (цифру 0) и букву O.
 - Пользуйтесь принятыми в SPICE буквенными масштабными множителями (**MEG** вместо **M** для обозначения множителя **1E+6** или **106**).
 - Не допускаются пробелы между значениями и буквенными множителями. Например, емкость должна быть задана как **1.0uF**, а не **1.0 uF**.
 - Проверьте правильность установки параметров всех устройств и источников.
 - Убедитесь в корректности задания коэффициента усиления всех зависимых источников тока и напряжения.
 - Временно исключите последовательно включенные конденсаторы или источники тока.
 - Временно исключите параллельно включенные катушки индуктивности или источники напряжения.
3. Увеличьте максимально допустимое число итераций в окне **Analog Options (ITL1)** до 300.
 4. Добавьте устройства **.NS (Nodeset)**, для того чтобы определить узловые напряжения. Если не определены начальные приближения узловых потенциалов, то можно воспользоваться этими элементами **.NS** для предварительной установки напряжения, которое будет использовано на первом шаге расчета рабочих точек. За более подробной информацией по использованию элемента **Nodeset** обратитесь к параграфу *Задание начальных условий* раздела *Подготовка и проведение моделирования*.
 5. Если введение элемента **.NS** не помогает сходимости, попробуйте задать начальные условия при помощи элемента **.IC**. В этом случае значения напряжений в узлах поддерживаются постоянными во время расчета рабочей точки и разблокируются при анализе переходных процессов.
 6. Включите опцию **Use Initial Conditions** (использовать начальные условия) диалогового окна **Analyses Setup**. Эта опция работает совместно с элементом **.IC** или с параметром **IC** элемента схемы. Отличие этой опции состоит в том, что при ее применении расчет рабочей точки не проводится, а установленные напряжения в узлах используются в качестве начальных условий для анализа переходных процессов. За более подробной информацией по определению начальных условий обращайтесь к соответствующему параграфу раздела *Подготовка и проведение моделирования*.
 7. Задайте параметры последовательных сопротивлений ваших моделей и увеличьте параметр **GMIN** в десять раз. Запретите использование начальных условий для полупроводниковых приборов, особенно для диодов.

Устранение ошибок, возникающих при анализе схемы по постоянному току

1. Проверьте правильность прорисовки схемы и связанность. Посмотрите общие ошибки, описанные в предыдущем разделе.
2. Увеличьте значение параметра **ITL2** в диалоговом окне **Analog Options** до 200. Это позволит программе анализа по постоянному току пройти большее количество итераций на каждом шаге до прерывания.
3. Увеличьте или уменьшите шаг приращения постоянного тока (**DC sweep step**). Если в модели устройства имеется точка разрыва функции (возможно между линейной областью и зоной насыщения модели), то увеличение шага позволит программе моделирования перескочить через разрыв (неопределенность) функции. Уменьшение шага приращения позволяет программе моделирования рассчитывать резкие перепады напряжений в точках разрыва функции.
4. Откажитесь от анализа схемы по постоянному току. Некоторые задачи, например, расчет гистерезиса, не могут быть разрешены посредством анализа по постоянному току. В таких случаях более эффективным может оказаться использование анализа переходных процессов, с изменением по линейному закону соответствующих источников питания.

Устранение ошибок, возникающих при анализе переходных процессов

- Проверьте правильность прорисовки схемы и связанность. Посмотрите общие ошибки, описанные в предыдущих разделах.
- В диалоговом окне **Analog Options** установите значение параметра **RELTOL** равным 0.01. При увеличении допуска до 0.001 (точность 0.1%) для сходимости потребуется меньшее число итераций и процесс моделирования завершится быстрее.
- В диалоговом окне **Analog Options** увеличьте значение параметра **ITL4** до 100. Это позволит программе производить большее число итераций на каждом временном шаге до прерывания. Увеличение этого параметра поможет избежать появления ошибок, подобных **Timestep too small**, улучшая сходимость и скорость моделирования.“
- Если позволяют уровни токов или напряжений, то уменьшите точность **ABSTOL/VNTOL**. Для анализа конкретной цепи, возможно, не потребуется разрешающей способности на уровне 1 мкВ или 1 пА. Устанавливаемое значение должно быть примерно на порядок меньшим, чем минимальный ожидаемый уровень напряжений или токов в схеме.
- Моделируйте схему максимально приближая ее к реальной. Добавляйте реальные паразитные элементы, особенно паразитные емкости и емкости переходов. Окружайте диоды сглаживающими RC цепочками. Модели готовых устройств, особенно мощных и высокочастотных, замещайте эквивалентными подсхемами.

- Увеличьте время нарастания/спада импульсных генераторов. Даже самые лучшие генераторы импульсов не могут переключаться мгновенно.
- Измените метод интегрирования на **Gear** (метод прямоугольников). Этот метод интегрирования требует большего времени моделирования, но работает, как правило, более стабильно, чем интегрирование по методу трапеций. Ступенчатое интегрирование полезно применять для моделирования схем генераторов и схем с обратными связями.

Предупреждения и сообщения об ошибках

Иногда программа моделирования выводит на экран сообщения с предупреждениями и сообщениями об ошибках. Предупреждения сохраняются в одноименном файле с расширением **SIM**, а сообщения об ошибках — в одноименном файле с расширением **ERR**.

Предупреждения (Warnings Messages)

Предупреждения, как правило, не прерывают процесс моделирования. Главным образом, они информируют о тех изменениях, которые должна сделать программа SPICE в цепи для корректного завершения процесса моделирования, например, изменение неправильно заданных или недостающих параметров.

Предупреждения интерпретатора SimCode могут включать информацию о нарушениях синхронизации (**tsetup**, **thold**, **trac**, **tw** и т. д.) или значительных перепадах напряжения в цепи питания цифровых элементов.

Примечание: Полноценные результаты моделирования генерируются даже при наличии предупреждений.

Сообщения об ошибках (Error Messages)

Сообщения об ошибках информируют о тех проблемах, которые программа SPICE не может разрешить сама, и которые приводят к прерыванию процесса моделирования. Наличие сообщений об ошибках говорит о том, что результаты моделирования сформированы быть не могут, и требуется обязательное их исправление.

Переменные и параметры модуля SPICE

Программа моделирования системы Protel 99 SE имеет доступ к набору внутренних параметров модуля SPICE. В этот набор входят такие параметры, как максимальное число итераций, температура, задержка распространения, шунтирующее сопротивление узла и т. д. (рис. 4.38).

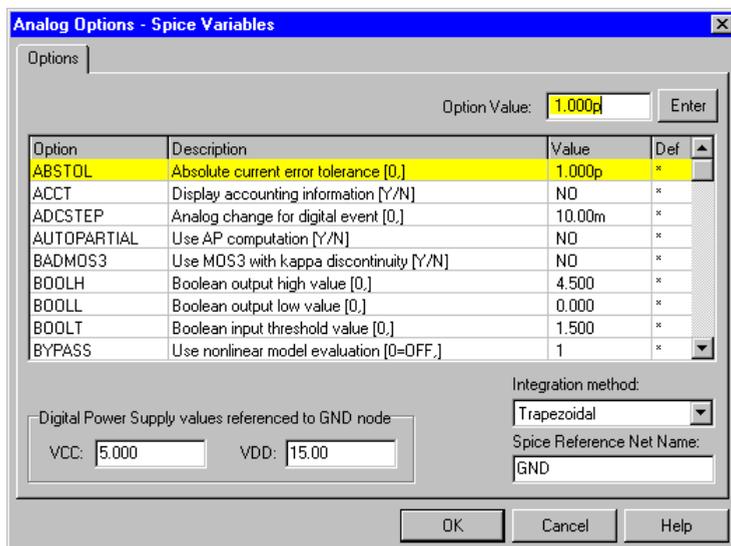


Рис. 4.38. Редактирование внутренних переменных SPICE.

Внутренние установочные параметры предназначены для опытных пользователей, изменять их следует только после внимательного изучения соответствующей литературы (в разделе *Обработка ошибок, возникающих при моделировании* рекомендуется делать эти изменения только при определенных условиях), когда пользователь в точности осознает свои действия.

Управление переменными и параметрами модуля SPICE осуществляется из диалогового окна **Analog Options — SPICE Variables**, вызываемого нажатием кнопки **Advanced**, расположенной в окне **Analyses Setup** (команда меню **Simulate » Analyses Setup**). Здесь задаются:

Метод интегрирования

В выпадающем списке **Integration Method** выбирается численный метод интегрирования. Метод трапеций (**Trapezoidal**), устанавливаемый по умолчанию, является относительно быстрым и точным методом, но при определенных условиях может работать неустойчиво. Метод прямоугольников (**Gear**) требует большего времени расчета, но работает более стабильно. Уменьшение шага разбиения позволяет получить более точный результат ценой еще большего увеличения времени интегрирования.

Источники питания цифровых компонентов и общий провод

Все цифровые компоненты, используемые программой моделирования, имеют скрытые выводы питания. Для серии микросхем 74xx (ТТЛ) это выводы **VCC** и **GND**, для серии 4000 (КМОП) — **VDD** и **GND**. Во время создания списка соединений все эти скрытые выводы автоматически соединяются между собой соответствующим обра-

зом, а подаваемые на них напряжения задаются в текстовых полях **VCC** и **VDD** в окне **Analog Options**. При необходимости эти напряжения можно изменить.

Для того, чтобы подключить схему к другим цепям питания, отличным от **VCC** или **VDD**, необходимо ввести в схему источник питания с соответствующим напряжением, сделать видимыми скрытые выводы питания компонентов и проводниками соединить их между собой.

Для проведения анализа переходных процессов, в случае если в качестве опорной используется цепь, отличная от общего провода **GND**, необходимо ввести название этой цепи в поле **Spice Reference Net Name**.

Опции SPICE

В таблице 4.43 приводится список переменных и параметров программы SPICE и описание их влияния на процесс моделирования. Чтобы изменить значение переменной SPICE, сначала необходимо выделить переменную, затем в поле **Option Value** ввести с клавиатуры ее новое значение, после чего нажать клавишу **ENTER**.

Таблица 4.43

Опция	Описание
ABSTOL	Абсолютная допустимая ошибка расчета токов в программе. Рекомендуется устанавливать $ABSTOL = RELTOL * (\text{Наименьшая величина тока в цепи})$. По умолчанию = 1 пА.
CHGTOL	Устанавливает нижнюю границу заряда емкости или магнитного потока индуктивности. Используется LTE алгоритмом управления временным шагом. По умолчанию = 1.0E-14 Кл.
DEFAD	Диффузионная площадь стока МОП транзистора. По умолчанию = 0.0 м ² .
DEFAS	Диффузионная площадь истока МОП транзистора. По умолчанию = 0.0 м ² .
DEFL	Длина канала МОП транзистора. По умолчанию = 100.0 мкм
DEFW	Ширина канала МОП транзистора. По умолчанию = 100.0 мкм
GMIN	Минимальная проводимость ветви цепи (максимальное сопротивление). Эта величина также устанавливает значение проводимости, помещаемой в схему параллельно р-п переходу. По умолчанию = 1.0E-12 См. Примечание: Для анализа многих цепей увеличение этого значения может помочь сходимости процесса анализа, но уменьшить точность вычислений.

ITL1	Максимальное количество итераций при расчете рабочих точек. По умолчанию = 100 итераций. Примечание: Для большинства цепей этот параметр может понадобиться увеличить до 500.
ITL2	Максимальное количество итераций при анализе передаточных функций по постоянному току. По умолчанию = 50 итераций. Примечание: Для некоторых цепей этот параметр может понадобиться увеличить до 200.
ITL3	Устанавливает минимальное количество итераций при анализе переходных процессов. По умолчанию = 4 итерации. Примечание: Данный параметр в SPICE3 не применяется, он необходим лишь для обеспечения совместимости с программой SPICE2 при экспорте списков соединений.
ITL4	Устанавливает максимальное количество итераций на каждом временном шаге при анализе переходных процессов. По умолчанию = 10 итераций. Примечание: Увеличение значения до 100 и более помогает избежать появления ошибок типа "timestep too small" (слишком маленький временной шаг), тем самым, улучшая сходимость и увеличивая скорость процесса моделирования.
ITL5	Устанавливает общее максимальное количество итераций при анализе переходных процессов. По умолчанию = 5000 итераций. Примечание: Данный параметр в SPICE3 не применяется, он необходим лишь для обеспечения совместимости с программой SPICE2 при экспорте списков соединений.
PIVREL	Устанавливает относительный коэффициент между самым большим значением содержимого столбца матрицы и допустимым ведущим элементом. Значение должно находиться между 0 и 1. По умолчанию = 1E-3. В численном алгоритме минимально допустимый ведущий элемент столбца определяется по формуле $EPSREL=AMAX1(PIVREL*MAXVAL,PIVTOL)$, где MAXVAL — максимальный элемент столбца, в котором ищется ведущий элемент.
PIVTOL	Устанавливает абсолютное минимальное значение элемента матрицы, принятого в качестве ведущего элемента. По умолчанию = 1.0E-13.
RELTOL	Максимально допустимая относительная ошибка при расчетах программой моделирования. Значение должно находиться между 0 и 1. По умолчанию = 0.001 (0.1%). При задании больших значений сокращается время моделирования, но ухудшается точность.

Таблица 4.43. Продолжение

TEMP	Устанавливает фактическую рабочую температуру схемы. Любое отклонение температуры от номинальной (TNOM) приведет к изменению в результатах моделирования. По умолчанию = 27°C. Примечание: параметр TEMP может быть переопределен заданием температуры любой температурно-зависимой модели устройства.
TNOM	Устанавливает номинальную температуру модели устройства. По умолчанию = 27°C. Примечание: параметр TNOM может быть переопределен заданием температуры любой температурно-зависимой модели устройства.
TRTOL	Используется LTE алгоритмом управления временным шагом. Является оценочным показателем, по которому SPICE переоценивает абсолютную ошибку округления. По умолчанию = 7.0
VNTOL	Определяет абсолютное значение ошибки при расчете напряжения. Рекомендуется устанавливать $VNTOL=RELTOL*$ (наименьшее значение напряжения цепи). По умолчанию = 1 мкВ.
BOOLL	Определяет выходное напряжение низкого уровня (логического нуля) в булевых выражениях для цифровых схем. По умолчанию = 0.0 В
BOOLH	Определяет выходное напряжение высокого уровня (логической единицы) в булевых выражениях для цифровых схем. По умолчанию = 4.5 В
BOOLT	Определяет пороговый уровень (уровень порога срабатывания логического элемента) в булевых выражениях для цифровых схем. По умолчанию = 1.5 В.
BADMOS3	Используется ранней версией модели MOS3 с разрывом функции типа "каппа". По умолчанию = NO (не использовать старую версию).
KEEPOPINFO	Сохраняет информацию о рабочих точках во время выполнения анализа по переменному току. Примечание: чрезвычайно полезная переменная при анализе больших схем, позволяющая экономить время при анализе частотных характеристик. По умолчанию = NO (расчет рабочих точек проводится каждый раз).
TRYTOCOMPACT	Применяется для LTRA модели. Если переменная определена, программа моделирования пытается сжать предисторию входных напряжений и токов LTRA линий передач. По умолчанию = NO (не сжимать).

Таблица 4.43. Продолжение

NOOPITER	Перейти непосредственно к пошаговому выполнению алгоритма GMIN. По умолчанию = NO (не переходить).
GMINSTEP	Задаёт число шагов при выполнении алгоритма GMIN. При нулевом значении этого параметра пошаговое выполнение GMIN алгоритма полностью отключается, при этом источником пошагового выполнения становится установленный по умолчанию программой моделирования алгоритм расчёта рабочих точек. По умолчанию = 10.
SRCSTEP	Задаёт число шагов алгоритма пошагового изменения источника для сходимости алгоритма расчёта рабочих точек. По умолчанию = 10.
ACCT	Выводит на экран статистику расчёта и время выполнения процесса моделирования. По умолчанию = NO (не выводить).
LIST	Выводит на экран полный список всех элементов схемы со всеми связями и значениями. По умолчанию = NO (не выводить).
OPTS	Выводит на экран полный список всех опций SPICE3 со значениями. По умолчанию = NO (не выводить).
BYPASS	Включает схему шунтирования для оценки нелинейности моделей. По умолчанию = 1 (включено).
MINBREAK	Определяет минимально допустимое время между точками останова программы. По умолчанию = 0 секунд (время устанавливается автоматически).
MAXOPALTER	Задаёт максимальное количество чередований аналоговый отсчёт/событие для сходимости расчёта рабочих точек. По умолчанию = 0.
MAXEVTITER	Задаёт максимальное количество итераций событий для сходимости расчёта рабочих точек. По умолчанию = 0.
NOOPALTER	Разрешает чередование при расчёте рабочих точек. По умолчанию = NO.
RAMPTIME	Управляет временем включения независимых источников, а также временем установлением начальных условий на конденсаторах и катушках индуктивности от нуля до заданного значения за определённый период времени. По умолчанию = 0.0 секунд.
CONVLIMIT	Отключает встроенный алгоритм сходимости, используемый в моделях компонентов. По умолчанию = NO.
CONVSTEP	Устанавливает границы относительного размера шага при выполнении расчёта рабочих точек для входов модели. По умолчанию = 0.25.
CONVABSSTEP	Устанавливает границы абсолютного размера шага при выполнении расчёта рабочих точек для входов модели. По умолчанию = 0.1.

Таблица 4.43. Продолжение

AUTOPARTIAL	Разрешает модулям XSPICE автоматическое вычисление частных производных. По умолчанию = NO.
PROMNS	Задаёт масштабный множитель, используемый для определения минимальной задержки распространения, если её действительное значение не определено в модели SimCode. По умолчанию = 0.5 (50% от типовой задержки распространения).
PROMXS	Задаёт масштабный множитель, используемый для определения максимальной задержки распространения, если её действительное значение не определено в модели SimCode. По умолчанию = 1.5 (150% от типовой задержки распространения).
TRANMNS	Задаёт масштабный множитель, используемый для определения минимального времени установления переходных процессов, если его действительное значение не определено в модели SimCode. По умолчанию = 0.5 (50% от типового времени переходного процесса).
TRANMXS	Задаёт масштабный множитель, используемый для определения максимального времени установления переходных процессов, если его действительное значение не определено в модели SimCode. По умолчанию = 1.5 (150% от типового времени переходного процесса).
LOADMNS	Масштабный множитель, определяющий минимальную входную нагрузку (максимальное входное сопротивление), если его действительное значение не определено в модели SimCode. По умолчанию = 1.5 (150% от типового входного сопротивления).
LOADMXS	Масштабный множитель, определяющий максимальную входную нагрузку (минимальное входное сопротивление), если его действительное значение не определено в модели SimCode. По умолчанию = 0.5 (50% от типового входного сопротивления).
DRIVEMNS	Масштабный множитель, определяющий минимальную нагрузку по выходу (максимальное выходное сопротивление), если его действительное значение не определено в модели SimCode. По умолчанию = 1.5 (150% от типового выходного сопротивления).
DRIVEMXS	Масштабный множитель, определяющий максимальную нагрузку по выходу (минимальное выходное сопротивление), если его действительное значение не определено в модели SimCode. По умолчанию = 0.5 (50% от типового выходного сопротивления).
CURRENTMNS	Масштабный множитель, определяющий минимальный ток источника питания (максимальное внутреннее сопротивление), если его действительное значение не определено в модели SimCode. По умолчанию = 1.5 (150% от типового внутреннего сопротивления).
CURRENTMXS	Масштабный множитель, определяющий максимальный ток источника питания (минимальное внутреннее сопротивление), если его действительное значение не определено в модели SimCode. По умолчанию = 0.5 (50% от типового внутреннего сопротивления).

Таблица 4.43. Продолжение

TPMNTYMX	Временная глобальная переменная перекрытия задержки распространения, указывающая на устройства SimCode (0 — по умолчанию, 1 — мин., 2 — типов., 3 — макс). По умолчанию = 0.
TTMNTYMX	Временная глобальная переменная перекрытия времени установления переходных процессов, указывающая на устройства SimCode (0 — по умолчанию, 1 — мин., 2 — типов., 3 — макс). По умолчанию = 0. transition time index on SimCode devices (0=default, 1=min, 2=typ, 3=max). Default=0.
LDMNTYMX	Временная глобальная переменная перекрытия нагрузки по входу, указывающая на устройства SimCode (0 — по умолчанию, 1 — мин., 2 — типов., 3 — макс). По умолчанию = 0.
DRVMNTYMX	Временная глобальная переменная перекрытия нагрузки по выходу, указывающая на устройства SimCode (0 — по умолчанию, 1 — мин., 2 — типов., 3 — макс). По умолчанию = 0.
IMNTYMX	Временная глобальная переменная перекрытия для тока источника питания, указывающая на устройства SimCode (0 — по умолчанию, 1 — мин., 2 — типов., 3 — макс). По умолчанию = 0.
SIMWARN	Задание ненулевого значения позволяет выводить на экран предупреждения интерпретатора SimCode во время выполнения процесса моделирования. Предупреждения интерпретатора SimCode могут содержать информацию о нарушениях синхронизации (tsetup, thold, tres, tw и т. д.) или указывать на падение напряжения источника питания ниже заданного. По умолчанию = 0.
RSHUNT	Омическое сопротивление резистора, подключаемого между каждым узлом схемы и землей, помогающего избежать появления ошибок типа "singular matrix" (вырожденная матрица). Вообще говоря, значение RSHUNT должно выбираться очень большим (порядка 1012 Ом). По умолчанию = 0 (нет шунтирующих резисторов). “
ADCSTEP	Минимальный размер шага, необходимый для регистрации событий на входе внутренних АЦП. По умолчанию = 0.01 В.

Моделирование цифровых схем

Программа моделирования схем системы Protel 99 SE — это программа моделирования смешанных сигналов. Это означает, что она может анализировать схемы, включающие как аналоговые, так и цифровые устройства. Процесс создания таких смешанных схем представляет собой всего лишь процесс размещения компонентов и соединения их проводниками. Каждый аналоговый элемент из библиотек, предназначенных для моделирования, ссылается на SPICE модель, а каждый цифровой компонент — на SimCode модель. Как только создание схемы полностью завершено, можно запускать процесс ее моделирования. Существует две библиотеки цифровых компонентов **74xx.lib** и **CMOS.lib**, расположенных в базе данных **Program Files\Design Explorer 99 SE\Library\Sch\Sim.ddb**.

Модели цифровых компонентов

Из-за сложности цифровых устройств в большинстве случаев нецелесообразно использовать при их моделировании стандартные, не управляемые событиями SPICE команды. По этой причине программа моделирования включает специальный описательный язык, позволяющий моделировать цифровые устройства с помощью расширенной, управляемой событиями версии программы XSPICE. Цифровые устройства, входящие в библиотеки моделирования, используют язык Digital SimCode™.

Digital SimCode является патентованным языком, поэтому устройства, созданные с его помощью, не совместимы с другими программами моделирования, а также не совместимы с цифровыми компонентами, созданными системой Protel для других программ моделирования.

Источники питания цифровых устройств и общий провод

Любое цифровое устройство имеет в своем условном графическом обозначении скрытые выходы питания (**VCC** для 74xx, **VDD** для КМОП устройств) и земли (**GND**), которые автоматически соединяются между собой во время создания списка соединений. Программа моделирования использует для имен таких цепей значения по умолчанию, поэтому в случае исключительно цифрового проектирования нет необходимости задавать источники для питания компонентов. Если схема помимо цифровых содержит какие-либо аналоговые компоненты, соединенные с питающими шинами **VCC** или **VDD**, например, нагрузочный резистор, тогда нужно включить в схему соответствующие источники питания (рис. 4.39).

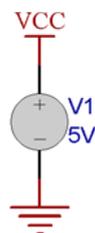


Рис 4.39. Организация питания цифровых компонентов для проведения смешанного аналого-цифрового моделирования

Значения **VCC** и **VDD**, используемые в процессе моделирования, и имя опорной цепи (по умолчанию **GND**) задаются в диалоговом окне **Analog Options**, которое вызывается с помощью кнопки **Advanced**, расположенной в диалоговом окне **Analyses Setup**.

Создание нового SimCode устройства

SimCode представляет собой Си-подобный описательный язык, используемый для задания характеристик и поведения устройства, модель которого необходимо построить. Он содержит функции для определения таких параметров, как задержки распространения, нагрузочные характеристики, мощность и т. д. Поведение устройства определяется с помощью таблиц истинности, математических функций и условных выражений, таких как **IF ... THEN**. Обзор всех элементов языка приведен в подразделе *Определение языка SimCode*, а подробное описание каждого элемента в отдельности — в подразделе *Синтаксис языка SimCode*.

Следующий пример описывает процесс создания собственной SimCode модели.

Пример создания условного графического обозначения и SimCode модели устройства 74LS74

Ниже приведено пошаговое описание процесса создания и подготовки к моделированию микросхемы 74LS74, содержащей два D-триггера, защелкивающиеся по положительному перепаду.

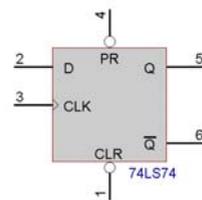


Рис. 4.40. Создание УГО компонента

Шаг 1 — создание условного графического обозначения

Первым шагом является создание в редакторе библиотек элементов схем условного графического обозначения устройства (рис. 4.40). Микросхема 74LS74 представляет собой двухсекционный элемент. **VCC** (вывод 14) и **GND** (вывод 7) определяются как скрытые выводы.

Подробное описание процесса создания символического обозначения приведено в разделе *Библиотеки элементов принципиальных схем* главы *Разработка принципиальных схем*.

Шаг 2 — Определение данных для связи с моделью

Следующий шаг — заполнение библиотечных текстовых полей (**Library Text Fields**). Эти данные связывают символическое обозначение с SimCode моделью.

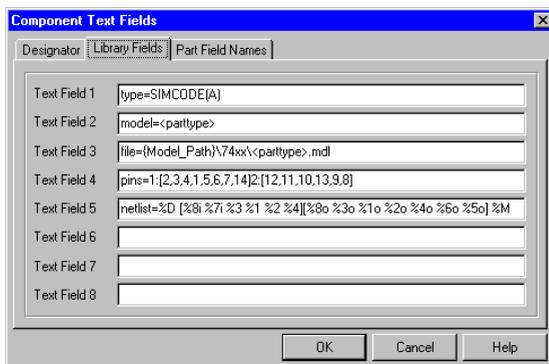


Рис. 4.41. Заполнение библиотечных полей для связи с моделью.

Использование библиотечных полей описано в подразделе *Создание собственных компонентов и подготовка их к моделированию* раздела *Компоненты и модели*. В полях 1, 2 и 3 введите тип компонента, имя модели и местоположение файла модели.

Теперь все готово для спецификации выводов (**Text Field 4**) и данных списка соединений, которые включают информацию о положении и номерах выводов (**Text Field 5**).

Text Field 4. Поскольку компонент имеет два элемента, строка в этом поле pins= будет состоять из двух частей:

```
pins=1:[ ... ]2:[ ... ]
```

Порядок перечисления выводов здесь не фиксирован, но для удобства они перечислены слева направо и сверху вниз для каждого элемента.

```
pins=1:[2,3,4,1,5,6,7,14]2:[12,11,10,13,9,8]
```

Text Field 5. Первый элемент этой строки — ссылка на позиционное обозначение (%D). Последний элемент — ссылка на модель (%M).

Между этими ссылками находится информация о положении и номерах выводов, которая называется также списком узлов. Синтаксис для цифровых и аналоговых компонентов различен. Для первых существует две части отдельно для входных и выходных узлов, каждая из которых заключается в квадратные скобки.

Узлы должны быть перечислены в том же порядке, что и выводы в **INPUTS** и **OUTPUTS** выражениях в SimCode. Если обратиться к примеру исходного файла для микросхемы 74LS74, то можно увидеть следующие строки для входных и выходных выражений:

```
INPUTS VCC, GND, PRE, DATA, CLK, CLR;
OUTPUTS VCC_LD, PRE_LD, DATA_LD, CLK_LD, CLR_LD, QN, Q;
(Синтаксис этих выражений будет позднее в данном примере.)
```

Самый простой способ сделать раскладку выводов — начертить таблицу с перечислением выводов в порядке, который требуется для SimCode модели, затем в соответствующей строке таблицы вписать положение этих выводов в списке выводов (**Text Field 4**), как показано в таблице 4.44.

Таблица 4.44

Входной вывод	Позиция вывода в списке выводов	Выходной вывод	Позиция вывода в списке выводов
VCC	8	VCC_LD	8
GND	7	PRE_LD	3
PRE	3	DATA_LD	1
DATA	1	CLK_LD	2
CLK	2	CLR_LD	4
CLR	4	QN	6
		Q	5

Отсюда получаем список узлов путем добавления символа "i" (input) после каждого входного вывода и символа "o" (output) после каждого выходного. В результате запись в поле **Text Field 5** приобретет следующий вид:

```
netlist=%D [%8i %7i %3 %1 %2 %4][%8o %3o %1o %2o %4o %6o %5o] %M
```

Шаг 3 — Создание промежуточного файла связи модели

Следующий шаг — создание файла связи модели (файла с расширением **MDL**), который связывает условное графическое обозначение с ASCII файлом SimCode модели. После того, как модель будет скомпилирована, созданный на этом шаге файл изменится таким образом, что уже будет ссылаться на скомпилированный файл SimCode модели.

В любом текстовом редакторе нужно создать новый файл с именем **74LS74.MDL**. Далее необходимо его сохранить в том месте, которое определено в поле **Text Field 3** для данного условного графического обозначения.

MDL файл включает одну строку **.MODEL** и необходимое количество строк комментариев (начинающихся с символа "*").

Синтаксис строки **.MODEL**:

```
.MODEL 74LS74 xsimcode(file="{MODEL_PATH}74LS74.TXT" func=ls74
```

- .MODEL** Объявляет выражение для модели.
- 74LS74** Имя модели (как в поле **Text Field 2**).
- xsimcode** Тип модели для цифровой SimCode модели.
- file=** Указывает на файл, содержащий цифровой SimCode код данного устройства. **{MODEL_PATH}** заменяется на имя папки моделей (**Models**), которая определена в файле **ADVSIM.INI**.
- func=** Определяет функцию цифрового SimCode кода данного устройства.
- data=** Содержит данные в ASCII коде для функции **READ_DATA** (необязательно).
- {mntymx}** Передает параметры цифровой модели (**Digital Model Parameters**) устройства в SimCode (должно в точности соответствовать приведенному).

Шаг 4 — Запись исходного кода для файла SimCode модели

Последний шаг — создание цифровой SimCode модели для устройства. Это можно сделать в любом текстовом редакторе в формате ASCII. Этот файл может иметь любое имя и расширение, но такое, чтобы оно совпадало с именем, заданным в параметре **file=** в файле с расширением **MDL**. Обычно имя содержит название описываемого устройства, например, **74LS74.TXT**. В одном файле можно разместить несколько SimCode моделей и ссылаться на них с помощью параметра **func=**.

Далее в этом разделе приводится пример SimCode кода для микросхемы 74LS74. Этот SimCode код можно скопировать непосредственно из файла справки через буфер обмена или ввести с помощью клавиатуры.

Правильность функционирования нового устройства рекомендуется проверять путем создания простой схемы и последующего ее тестирования. Заметим, что для одно-

значного определения ошибок функционирования во время тестирования в схеме должно присутствовать не более одного нового устройства.

При запуске программы моделирования исходный код модели будет автоматически скомпилирован и записан в текстовый файл **SIMLIST.TXT**, который будет сохранен в директории, где находится проект моделируемой схемы. Этот файл также содержит листинг процесса выполнения исходного кода модели. Затем можно надлежащим образом переработать исходный SimCode код модели и продолжать тестирование до окончательной отладки данной модели.

Шаг 5 — Создание скомпилированного файла SimCode модели

После того, как SimCode код будет успешно скомпилирован, можно извлечь информацию о скомпилированной модели из файла **SIMLIST.TXT** и создать файл скомпилированной модели (например, **74LS74.SCB**). Существует возможность сохранить несколько моделей в одном файле, но необходимо присвоить параметру **file=** в файле MDL значение, совпадающее с именем файла библиотеки скомпилированных SimCode моделей. По умолчанию эти модели хранятся в папке **\Program Files\Design Explorer 99 SE\Library\Sim**.

SimCode для микросхемы 74LS74

Далее следует описание всех частей исходного SimCode кода, описывающего поведение устройства 74LS74.

Секция 1 — Идентификация функции SimCode

Строка **# ls74 source** идентифицирует начало исходного текста SimCode функции для 74LS74.

Секция 2 — Описание данных

Эта секция содержит описание выводов и переменных.

Команда **INPUTS** объявляет имена входных выводов, а также выводов **VCC** и **GND**. Порядок перечисления выводов в этом выражении и в библиотечном текстовом поле 5 (**Library Text Field 5**) символьного обозначения должны совпадать.

Команда **OUTPUTS** объявляет имена выходных выводов. Заметим, что сюда также включаются и входные выводы, но с суффиксом **_LD**. Это делается для того, чтобы устройство могло обеспечить обратную нагрузку на управляющую цепь. В это выражение включен вывод **VCC**, но нет вывода **GND**. Порядок перечисления выводов в этом выражении и в поле **Library Text Field 5** символьного обозначения должны совпадать.

Команда **PWR_GND_PINS** объявляет выводы питания и заземления и определяет уровни напряжения на них для дальнейшего использования в этом SimCode коде.

Секция 3 — Инициализация SimCode функции

Секция **IF (init_sim) THEN** выполняется только один раз в начале процесса моделирования. Здесь устанавливаются характеристики устройства, которые не должны подвергаться изменениям при внешних воздействиях, например справочные данные. Здесь

также должны быть указаны наиболее вероятные состояния на выходах устройства. Секция завершается командой **EXIT**.

Секция 4 — Выражения **LOAD** и **DRIVE**

Эти выражения используются для задания нагрузочной способности по входу и выходу данного устройства.

Секция 5 — Выполняемые функции устройства

Эта часть может очень сильно отличаться у разных устройств. В данном примере использована команда **EXT_TABLE**. В других моделях устройств могут применяться **IF...THEN**, **STATE_BIT**, **NUMBER**, а также другие команды, определяющие логические функции устройства.

Секция 6 — Проверка на несоответствия в настройках устройства

Здесь производится проверка на несоответствия в настройках, которые в реальных условиях могут стать причиной неправильного функционирования данного устройства. Если эта функция включена, в процессе моделирования это устройство может работать, но будет выводиться разного рода предупреждения.

Секция 7 — Задержки по выходу/Передача событий

Команды **DELAY** ставятся в конце SimCode кода функции. Они передают события в программу моделирования, то есть сообщают об изменениях в состоянии устройства, и когда эти события будут обработаны, выполнение процесса моделирования продолжается. Каждому выходу устройства, в соответствии с справочными данными, задающим входным сигналом и выполняемыми функциями, присваивается некоторый временной параметр — задержка распространения.

```
//=====
1
# ls74 source
//1/2- 74LS74 D flip-flop Digital Simcode Model
//typical prop delay values from TI 1981 2nd edition data book
//=====

2
INPUTS VCC, GND, PRE, DATA, CLK, CLR;
OUTPUTS VCC_LD, PRE_LD, DATA_LD, CLK_LD, CLR_LD, QN, Q;
INTEGERS tblIndex;
REALS tplh_val, tphl_val, ts_val, th_val, trec_val, tt_val, temp_tp,
      clk_twl, clk_twh, pre_clr_twl, ril_val, rih_val, ricc_val;

PWR_GND_PINS(VCC,GND); //set pwr_param and gnd_param values
SUPPLY_MIN_MAX(4.75,5.25); //test for min supply=4.75 and max supply=5.25
VOL_VOH_MIN(0.2,-0.4,0.1); //vol_param=gnd_param+0.2,voh_param=pwr_param-
0.4
VIL VIH VALUE(1.25,1.35); //set input threshold values: vil and vih
IO_PAIRS(PRE:PRE_LD, DATA:DATA_LD, CLK:CLK_LD, CLR:CLR_LD);

3
IF (init_sim) THEN
  BEGIN //select prop delay, setup, hold, and width times
```

```

//MESSAGE("time\t\tPRE\tCLR\tCLK\tDATA\tQ\tQN"); //debug

//NOTE: both ttlh and tthl are the same value
tt_val= (MIN_TYP_MAX(tt_param: NULL, 5n, NULL));

temp_tp= (PWL_TABLE(sim_temp: -75, -5n, 125, 5n)); //tp temperature
affect
tplh_val= (MIN_TYP_MAX(tp_param: NULL, 14n, 25n) + temp_tp);
tphl_val= (MIN_TYP_MAX(tp_param: NULL, 20n, 40n) + temp_tp);

ts_val= (20n);
th_val= (5n);
trec_val= (5n);
clk_twl= (25n); //not specified - derived from fmax
clk_twh= (25n);
pre_clr_twl= (20n);

//LS stdout drive IOL max=8mA @ VOL typ=0.35V:rol_param=0.35V/8mA=43.75
//LS stdout drive IOL max=8mA @ VOL max=0.5V: rol_param=0.5V/8mA=62.5
rol_param= (MIN_TYP_MAX(drv_param: 62.5, 43.75, NULL));

//LS stdout drive IOS min=20mA @ VCC max=5.25V:
roh_param=5.25V/20mA=262.5
//LS stdout drive IOS max=100mA @ VCC
max=5.25V:roh_param=5.25V/100mA=52.5
roh_param= (MIN_TYP_MAX(drv_param: 262.5, NULL, 52.5));

//LS input load IIH max=20uA @ Vin=2.7V: ril= (2.7-vol_param)/20uA=125k
ril_val= (MIN_TYP_MAX(ld_param: NULL, NULL, 125k));
//LS input load IIL max=-0.4mA @ Vin=0.4V:rih= (voh_param-
0.4)/0.4mA=10.5k
rih_val= (MIN_TYP_MAX(ld_param: NULL, NULL, 10.5k));

//Icc @ 5V: 2500= 4mA/2 typical, 1250= 8mA/2 max
ricc_val= (MIN_TYP_MAX(i_param: NULL, 2500, 1250));

STATE Q = ONE; // initialize output states
STATE QN = ZERO;
EXIT;
END;

4
DRIVE Q QN = (v0=vol_param,v1=voh_param,ttlh=tt_val,tthl=tt_val);
LOAD PRE_LD DATA_LD CLK_LD CLR_LD =
(v0=vol_param,r0=ril_val,v1=voh_param,r1=rih_val,io=1e9,t=1p);

5
EXT_TABLE tblIndex
PRE CLR CLK DATA Q QN
0 1 X X H L
1 0 X X L H
0 0 X X H H
1 1 ^ X DATA ~DATA
1 1 X X Q ~Q;

//MESSAGE("%fs\t%d\t%d\t%d\t%d\t%d",present_time,PRE,CLR,CLK,DATA,Q,QN);
LOAD VCC_LD = (v0=gnd_param,r0=ricc_val,t=1p);

```

```

6
IF (warn_param) THEN
  BEGIN
    IF (PRE && CLR) THEN
      BEGIN
        SETUP_HOLD(CLK=LH DATA Ts=ts_val Th=th_val "CLK->DATA");
        RECOVER(CLK=LH PRE CLR Trec=trec_val "CLK->PRE or CLR");
        WIDTH(CLK Twl=clk_twl Twh=clk_twh "CLK");
        WIDTH(PRE CLR Twl=pre_clr_twl "PRE or CLR");
      END;
    END;
  END;

DELAY Q QN =
  CASE (TRAN_LH) : tphl_val
  CASE (TRAN_HL) : tphl_val
END;
EXIT;

```

Определение языка SimCode

В данном разделе приводится подробное описание элементов языка SimCode.

Таблица 4.45

INPUTS	Входные выводы (выводы, управляющие работой схемы)
OUTPUTS	Выходные выводы (выводы, связанные с источниками сигналов и нагрузкой)
INTEGERS	Переменные или массивы с элементами целого типа
REALS	Переменные или массивы с элементами вещественного типа
PWR_GND_PINS	Выводы питания и заземления, а также номиналы напряжений питания
IO_PAIRS	Соединения входных/выходных выводов для входной нагрузки

Функции настройки устройства

Эти функции используются для настройки определенных характеристик выводов устройства.

Таблица 4.46

VIL_VIH_VALUE	Абсолютные значения VIL и VIH
VIL_VIH_PERCENT	Значения VIL и VIH в процентах от напряжения питания
VOL_VOH_MIN	Значения VOL и VOH в зависимости от питания и заземления

Функции тестирования устройства

Эти функции используются для проверки противоречий в настройках устройства, которые могут иметь место в схеме. Однако, эти противоречия могут и не повлиять на выполняемые во время моделирования устройства функции, то есть оно может продолжать работать в процессе моделирования даже при наличии таких противоречий. Поэтому, чтобы знать об их присутствии нужно включить функцию выдачи предупреждений.

Таблица 4.47

SUPPLY_MIN_MAX	Проверка на выход за границы диапазона разрешенных значений напряжения на выводах питания
RECOVER	Проверка времени установления сигнала на входе
SETUP_HOLD	Проверка времени установления и удержания сигнала на входе
WIDTH	Проверка минимальной ширины импульса на входе
FREQUENCY(FMAX)	Проверка на выход за границы диапазона разрешенных значений частоты на входе

Функции выходных выводов

Эти функции используются для программирования выходных выводов устройства.

Таблица 4.48

STATE	Установка выходов в объявленное логическое состояние
STATE_BIT	Установка выходов в двоичные взвешенные логические состояния
LEVEL	Установка уровня выходного состояния
STRENGTH	Установка мощности состояния выхода
TABLE	Установка логических состояний выходов на основе таблицы истинности
EXT_TABLE	Установка логических состояний выходов на основе расширенной таблицы истинности
LOAD	Объявление нагрузочных характеристик входных выводов
DRIVE	Объявление нагрузочных характеристик выходных выводов
DELAY	Установка задержки распространения сигнала до указанного выхода
NO_CHANGE	Установка вывода в неизменное состояние
EVENT	Принудительная выдача цифрового события

Математические функции и операторы

Для манипулирования с данными и проверки условий, влияющих на выполнение программы, в программе выражения с различными операторами и функциями. Выражения всегда записываются в круглых скобках. Операторы в выражении выполняются слева направо, начиная с самых внутренних скобок.

Таблица 4.49

Операторы	+, -, *, /, ~, !, &&, , ^, &, , >>, <<, >, <, =, !=, >=, <=
Математические функции	POW, ABS, SQRT, EXP, LOG, LOG10, SIN, COS, TAN, ASIN, ACOS, ATAN, HSIN, H COS, HTAN

Выражения

Таблица 4.50

PARAM_SET	Определяет, были ли предварительно установлены какие-либо SimCode параметры
PWL_TABLE	Возвращает значение из интерполяционной поисковой таблицы
SELECT_VALUE	Возвращает значение из простой поисковой таблицы
MIN_TYP_MAX	Возвращает значение из MIN_TYP_MAX поисковой таблицы
NUMBER	Возвращает число, полученное на основе двоичных взвешенных состояний выводов
VALUE	Возвращает состояние указанного вывода
CHANGE_TIME	Возвращает время последнего изменения состояния указанного вывода
WIDTH_TIME	Возвращает ширину последнего импульса сигнала на указанном выводе
INSTANCE	Проверяет наличие указанного экземпляра устройства
CHANGED_xx	Проверяет, изменилось ли состояние указанного вывода
READ_DATA	Читает данные из ASCII файла в массивы

Программное управление

Эти выражения применяются для управления процессом выполнением программы.

Таблица 4.51

# xxxx source	Определяет начало SimCode кода функции
IF ... THEN	Задаёт условный переход
WHILE ... DO	Задаёт циклическое выполнение, пока условие истинно
GOTO	Определяет безусловный переход
GOSUB	Выполняет вызов подпрограммы
RETURN	Выполняет возврат из подпрограммы
EXIT	Определяет конец программы

Вывод текста

Эти команды используются для вывода сообщений во время моделирования и отладки.

Таблица 4.52

PROMPT	Приостановка процесса моделирования и вывод сообщения
MESSAGE	Вывод сообщения без остановки

Отладка

Эти команды управляют пошаговым выполнением программы в режиме отладки.

Таблица 4.53

STEP_ON	Включение режима пошагового выполнения
STEP_OFF	Выключение режима пошагового выполнения

Синтаксис языка SimCode

В данном разделе приведено подробное описание всех элементов языка SimCode. При описании синтаксиса здесь используются следующие стили:

- italics* зарезервированные слова или важное замечание
- <> значение/переменная/вывод/выражение
- [] необязательный параметр
- { }|{ } выбор (возможен выбор одного из этих параметров)

xxxx source

Определяет начало SimCode кода функции.

Основная форма

```
# <func name> source
```

Параметры

<func name> Имя SimCode функции.

Применение

Это выражение позволяет идентифицировать SimCode функцию, после чего она может быть использована для моделирования устройства. Каждая функция должна начинаться с этого выражения.

Замечания

Ядро моделирования программы SPICE способно читать как исходный, так и скомпилированный коды моделей. Ключевое слово source сообщает программе моделирования, что это исходный код, который она автоматически скомпилирует при запуске процесса моделирования. Скомпилированный код будет помещен в текстовый ASCII файл **SIMLIST.TXT** в той же директории, где расположен проект схемы, использующей данное устройство.

Пример

```
//=====
# MyDevice source
//=====
INPUTS VCC, GND, IN1, IN2
OUTPUTS VCC_LD, IN1_LD, IN2_LD, OUT
.
.
.
EXIT
```

CHANGE_TIME

Возвращает время последнего изменения состояния указанного вывода.

Основная форма

```
CHANGE_TIME (<pin>)
```

Параметры

<pin> Имя входного или выходного вывода.

Применение

Эта функция возвращает вещественное значение, показывающее время последнего изменения состояний указанного входного или выходного вывода.

Пример

```
T1 = (CHANGE_TIME(INA));
```

CHANGED_хх

Проверяет изменилось ли состояние указанного вывода.

Основная форма

```
CHANGED_хх(<pin> [{<}&|{<=&}|{>}&|{>=&} <var/time/value>])
```

Параметры

<pin> Имя входного или выходного вывода.
<var/time/value> Элемент, с которым сравнивается <pin>.

Применение

Эта функция используется для определения того, изменилось ли состояние указанного вывода. Часть ключевого слова `хх` может быть опущена, что устанавливает проверку всех типов изменений, или может принимать одно из следующих значений:

LH, LX, HL, HX, XL, XH, LZ, ZL, ZH, ZX, HZ, XZ

что устанавливает проверку специфичных типов изменений. Необязательные параметры, операторы сравнения и `<var/time/value>`, должны быть включены в выражение, если нужно сделать проверку более специфичных изменений. При их отсутствии функция возвратит 1, если состояние вывода изменилось на текущем шаге моделирования.

Пример

```
IF (CHANGED_LH(CLK)) THEN ...  
IF (CHANGED(DATA < 10n)) THEN ...
```

DELAY

Устанавливает задержку распространения сигнала до указанных выходов.

Основная форма 1

```
DELAY <output> [<output> ...] = <delay>;
```

Основная форма 2

```
DELAY <output> [<output> ...] =  
  CASE (<conditional exp>) : <delay>  
  CASE (<conditional exp>) : <delay>  
  [CASE (<conditional exp>) : <delay> ...]  
END;
```

Параметры

<output> Имя выходного вывода или переменная, содержащая его индекс.
<conditional exp> Условное выражение, определяющее какая величина задержки будет использована.
<delay> Задержка распространения до выходного вывода.

Применение

Команда **DELAY** выполняется для всех перечисленных в ней выводов и определяет задержку распространения для каждого вывода, который изменяет свой уровень. Опция **CASE** позволяет определить более одного значения задержки. Условие **<conditional exp>** устанавливает какое из значений задержек будет использовано. Если задержка установлена для вывода, состояние которого не изменяется, тогда этот вывод будет помечен флагом **NO_CHANGE**, и задержка для него не будет определена. Параметр **<delay>** может быть константой, переменной или выражением вещественного типа.

Замечания

Команда **DELAY** должна выполняться только для выходных выводов, то есть для выводов, объявленных в операторе **OUTPUTS** и отсутствующих в операторах **LOAD** и **NO_CHANGE**. Значения задержек устанавливаются в порядке перечисления их в команде **DELAY** (то есть первым устанавливается вывод, находящийся первым в списке). Условия **<conditional exp>** вычисляются также в порядке следования в команде до тех пор, пока какое-либо из условий не примет значение **TRUE**. После этого связанное с таким условием значение задержки будет установлено для соответствующего данной команде выходного вывода. Если при использовании опции **CASE** ни одно из условий **<conditional exp>** не принимает значение **TRUE**, то установится значение задержки, связанное с последним оператором **CASE**.

В дополнение к стандартным функциям, которые могут быть использованы в **<conditional exp>**, для этих целей существует еще ряд выражений, применимых исключительно к выходным выводам:

TRAN_LH	low-to-high	(из низкого в высокий)
TRAN_LX	low-to-other	(из низкого в любой другой)
TRAN_HL	high-to-low	(из высокого в низкий)
TRAN_HX	high-to-other	(из высокого в любой другой)
TRAN_HZ	high-to-tristate	(из высокого в высокоомный)
TRAN_XL	other-to-low	(в низкий)
TRAN_XH	other-to-high	(в высокий)
TRAN_LZ	low-to-tristate	(из низкого в высокоомный)
TRAN_ZL	tristate-to-low	(из высокоомного в низкий)
TRAN_ZH	tristate-to-high	(из высокоомного в высокий)
TRAN_ZX	tristate-to-other	(из высокоомного в любой другой)
TRAN_XZ	other-to-tristate	(в высокоомный)
TRAN_XX	other-to-different	(любое изменение уровня)

Если значение задержки меньше или равно нулю, то на экран будет выдано сообщение об ошибке времени выполнения (**run-time error**). Выходные выводы в команде **DELAY** могут быть заданы посредством их имен или целых переменных, которые содержат индекс выходного вывода. Однако, имена и переменные одновременно не могут быть использованы в одной команде.

Примеры

```
DELAY Q1 Q2 Q3 Q4 = 10n;
```

```

DELAY Q QN =
  CASE (TRAN_LH) : tphl_val
  CASE (TRAN_HL) : tphl_val
END;
data = (E0_1 && (CHANGED(D0) || CHANGED(D1)));
DELAY Q1 Q0 =
  CASE (data && TRAN_LH) : tphl_D_Q
  CASE (data && TRAN_HL) : tphl_D_Q
  CASE (TRAN_LH) : tphl_E_Q
  CASE (TRAN_HL) : tphl_E_Q
END;

```

В этом примере если data принимает не нулевое значение и уровень **Q1** изменяется от высокого к низкому, то для **Q1** будет установлено значение задержки равное **tphl_D_Q**. Если уровень **Q0** изменится от низкого к высокому, тогда для **Q0** будет установлено значение задержки равное **tphl_D_Q**.

DRIVE

Объявляет выходные характеристики выходов устройства.

Основная форма

```

DRIVE <output> [<output> ...] =
  (v0=<value> v1=<value> ttlh=<value> tthl=<value>);

```

Параметры

<output>	Имя выходного вывода или переменная, содержащая его индекс.
<value>	Константа или переменная вещественного типа.
v0	VOL для выходного вывода.
v1	VON для выходного вывода.
ttlh	Время изменения уровня от низкого к высокому.
tthl	Время изменения уровня от высокого к низкому.

Применение

Команда **DRIVE** используется для объявления выходных характеристик выходных выводов устройства. При низком уровне на выходном выводе на него через сопротивление **rol_param** подается напряжение, значение которого равно **v0**. При высоком уровне через сопротивление **roh_param** подается напряжение **v1**. Время изменения уровня от низкого к высокому для данного вывода устанавливается равным **ttlh**, а время изменения уровня от высокого к низкому — **tthl**.

Замечания

Имена выводов и индексные переменные одновременно не могут быть использованы в одной команде **DRIVE**.

Значение **rol_param** должно быть получено из справочных данных для **VOL**. Это значение представляет общее сопротивление насыщения соединенной с землей части выходного каскада на данном выходе устройства. Например, стандартный выход

микросхемы серии LS в состоянии логического нуля выдает ток 8 мА, а напряжение не превышает 0.5 В и обычно равно примерно 0.35 В. Таким образом:

Типичное значение для низкого уровня: $rol_param = VOL_{typ} / IOL_{max}$
 $rol_param = 0.35V / 8mA$
 $rol_param = 43.75\ ohms$

Минимальное значение для низкого уровня: $rol_param = VOL_{max} / IOL_{max}$
 $rol_param = 0.5V / 8mA$
 $rol_param = 62.5\ ohms$

Значение **roh_param** должно быть получено из справочных данных для **IOS**, если они доступны. Это значение представляет общее сопротивление насыщения соединенной с питанием части выходного каскада на данном выходе устройства. Например, стандартный выход микросхемы серии LS в состоянии логической единицы, в режиме короткого замыкания на землю, при $V_{cc} = 5.25V$ выдает ток от 20 мА до 100 мА. Таким образом:

Минимальное значение для высокого уровня: $roh_param = VCC_{max} / IOS_{min}$
 $roh_param = 5.25V / 20mA$
 $roh_param = 262.5\ ohms$

Максимальное значение для высокого уровня: $roh_param = VCC_{max} / IOS_{max}$
 $roh_param = 5.25V / 100mA$
 $roh_param = 52.5\ ohms$

Пример

```
rol_param = (MIN_TYP_MAX(drv_param: 62.5, 43.75, NULL);  
roh_param = (MIN_TYP_MAX(drv_param: 262.5, NULL, 52.5);  
DRIVE Q QN = (v0=vol_param,v1=voh_param,ttlh=ttlh_val,  
tthl=tthl_val);
```

Смотрите также:

LOAD

EVENT

Вызывает передачу в программу цифрового события.

Основная форма

```
EVENT = (<time>){<expression>}
```

Параметры

<time> Время, когда событие должно произойти.

<expression> Выражение, показывающее время, когда событие должно произойти.

Применение

В большинстве случаев цифровое событие посылается при изменении состояния одного или более выводов, перечисленных в операторе **INPUT** данной SimCode модели. При обработке события вызывается и запускается соответствующая ему SimCode процедура. Описываемая команда позволяет SimCode модели послать цифровое событие в нужное время, определяемое параметром **<time>**. Цифровое событие будет

послано, если установленное в команде время больше, чем время, прошедшее с начала моделирования и сохраняемое в переменной `present_time`. Если с помощью одного оператора **EVENT** посылается более одного события, то будет использовано только самое позднее время. Таким образом, эта функция позволяет создавать простые модели синхронно срабатывающих устройств.

Замечания

Если цифровое событие для какой-либо SimCode модели происходит до момента времени, указанного в команде **EVENT**, то в нужное время (то есть в момент времени `<time>`) оно посылается повторно. Например, 1) текущее время моделирования равно 1 мкс, 2) SimCode модель устанавливает **EVENT** = 2 мкс и 3) входной вывод в модели изменяет свое состояние в 1.5 мкс, тогда событие будет послано повторно в 2мкс.

Пример

```
EVENT = (present_time + 1e-6); //return in lus
```

EXIT

Завершает выполнение SimCode модели.

Основная форма

```
EXIT;
```

Применение

Оператор **EXIT** используется для завершения выполнения SimCode кода.

Замечания

Как правило, это последняя строка в SimCode модели, но она также может быть размещена в любом другом месте для прерывания выполнения оставшейся части SimCode кода.

EXT_TABLE

Устанавливает логические состояния выходов с помощью расширенной таблицы истинности.

Основная форма

```
EXT_TABLE <line>
<input pin> [<input pin> ...] <output pin> [<output pin> ...]
<input state> [<input state> ...]
                <output state> [<output state> ...];
```

Параметры

<code><line></code>	Переменная, в которой сохраняется номер строки, используемой в таблице
<code><input pin></code>	Имя входного вывода
<code><output pin></code>	Имя выходного вывода
<code><input state></code>	Состояние отдельных входов
<code><output state></code>	Состояние отдельных выходов, основанное на входных условиях

Применение

Команда **EXT_TABLE** является функцией расширенной таблицы истинности, которая используется для установки уровня и мощности перечисленных в ней выходов. Допустимые значения для входов:

- 0 низкий уровень (входное напряжение \leq *vil_param*)
- 1 высокий уровень (входное напряжение \geq *vih_param*)
- ^ перепад из низкого в высокий
- v перепад из высокого в низкий
- X значение входного напряжения не важно

Допустимые значения для выходов:

- L **ZERO** — логический ноль (уровень выходного сигнала = *vol_param*).
- H **ONE** — логическая единица (уровень выходного сигнала = *voh_param*).
- Z **UNKNOWN** — высокоомное состояние (уровень выходного сигнала = *v3s_param*).

Для задания состояний выходов также допустимо с именами входных и/или выходных выводов применять специальные префиксы:

- Состояние, идентичное предыдущему.
- ~ Состояние, противоположное (инверсное) текущему состоянию.
- ~ Состояние, противоположное (инверсное) предыдущему состоянию.

За символом состояния выхода может следовать двоеточие и символ, показывающий мощность:

- s **STRONG** (установка выхода в *rol_param* для состояния L (0) и в *roh_param* для состояния H (1)).
- z **HI_IMPEDANCE** (установка выхода в *r3s_param*).

Если символ мощности после состояния выхода отсутствует, то для состояний L и H принимается значение **STRONG**, а для высокоомного состояния Z — **HI_IMPEDANCE**.

Замечания

Строки проверяются последовательно сверху вниз пока не встретятся входные условия. Выходы установятся для первой строки, которая соответствует входным условиям. Переменная **<line>** примет значение равно номеру использованной при этом строки. Если при проверке таблицы ни одного соответствия не было найдено, то переменная **<line>** будет равна нулю. Имена выводов, используемых для определения состояния выходов, не обязательно должны присутствовать в заголовке таблицы. В отличие от оператора **TABLE** переменные для входов не допускаются.

Пример

```
EXT_TABLE tblIndex
PRE      CLR   CLK   DATA  Q      QN
0        1    X    X      H      L
1        0    X    X      L      H
0        0    X    X      H      H
1        1    ^    X      DATA  ~DATA
1        1    X    X      Q       ~Q;
```

Этот пример представляет расширенную таблицу истинности для D-триггера микросхемы 74LS74. Если состояния входных выводов **PRE**, **CLR** и **DATA** равны **H (1)** (то есть $\geq \text{vih_param}$), а сигнал на **CLK** переходит из **L** в **H**, то **Q** устанавливается в **H (1)** (voh_param) и **STRONG (roh_param)**, **QN** — в **L (0)** (vol_param) и **STRONG (rol_param)** и **tblIndex** становится равным 4.

Смотрите также

REALS, STATE, STATE_BIT, TABLE

FREQUENCY (FMAX)

Проверяет частоту входных сигналов на предмет выхода ее за пределы установленного диапазона.

Основная форма

```
FREQUENCY(<input> [<input>...] MIN=<frequency> MAX=<frequency>
["<message>"])
```

Параметры

<input> Имя тестируемого входного вывода или переменная, содержащая его индекс.

MIN Минимальная допустимая частота на тестируемом выводе.

MAX Максимальная допустимая частота на тестируемом выводе.

<message> Текстовая строка, выводимая при выходе за границы установленного диапазона.

Применение

Функция **FREQUENCY** сравнивает период сигнала на входе **<input>** (то есть время между соседними перепадами из **L (0)** в **H (1)** с обратной величиной значения в поле **<frequency>** (1/частота). Если этот период меньше обратной величины значения максимальной частоты или больше обратной величины значения минимальной частоты, тогда на экран выводится предупреждение (**WARNING**). В описываемую команду может быть добавлено необязательное поле **<message>**, которое в этом случае будет выдано на экран в качестве предупреждения.

Замечания

В этой команде должны быть использованы параметры из справочной документации на моделируемое устройство. В команду **FREQUENCY** одновременно могут быть включены и имена выводов, и индексные переменные. Заметим, что предупреждение выводится только для первой найденной ошибки.

Пример

```
FREQUENCY(CLK MAX=10MEG "CLK"); //check fmax only
```

GOSUB

Вызывает подпрограмму в SimCode модели.

Основная форма

```
GOSUB <label>;
```

Параметры

<label> Место в SimCode коде, откуда программа продолжит свое выполнение.

Применение

Команда **GOSUB** используется для изменения последовательности выполнения SimCode кода. Однако, в отличие от оператора **GOTO**, после того, как в вызванной части программы встретится оператор **RETURN**, выполнение программы продолжится с инструкции, следующей за **GOSUB**.

Пример

```
GOSUB Shift_Left;
.
.
.
Exit;
Shift_Left:
.
.
.
RETURN;
```

Смотрите также

RETURN

GOTO

Осуществляет переход в новое место SimCode кода.

Основная форма

```
GOTO <label>;
```

Параметры

<label> Место в SimCode кода, откуда программа продолжит свое выполнение.

Применение

Команда **GOTO** используется для изменения последовательности выполнения SimCode кода.

Замечания

Программа продолжит свое выполнение с места, где в SimCode коде находится метка <label>:. Первый символ метки <label> должен быть буквенным, за которым следует некоторое количество алфавитно-цифровых символов или символ подчеркивания (_). Сразу же за меткой должен идти символ двоеточия (:).

Пример

```
GOTO Shutdown;
```

```
.  
. .  
Shutdown:  
. .  
Exit;
```

Смотрите также

GOSUB, IF ... THEN

IF ... THEN

Выполняет некоторые действия согласно условию.

Основная форма 1

```
IF (<expression>) THEN BEGIN ... [ELSE ...] END;
```

Основная форма 2

```
IF (<expression>) THEN GOTO <label>;
```

Параметры

<expression> Какое-нибудь выражение, результат вычисления которого принимает значение **ИСТИНА** или **ЛОЖЬ**.

<label> Место в программе, с которого продолжится выполнение.

Применение

Оператор **IF ... THEN** используется для управления выполнением программы, основанного на том, какое значение, **ИСТИНА** или **ЛОЖЬ**, принимает выражение <expression>. Допускается вложение друг в друга несколько таких операторов.

Замечания

Если при использовании конструкции **BEGIN ... ELSE ... END** выражение <expression> принимает значение **ИСТИНА**, то выполняются операторы, следующие за словом **BEGIN**, и пропускаются все операторы между **ELSE** и **END**. Если выражение <expression> ложно и существует секция **ELSE**, тогда выполняются операторы этой секции, при отсутствии же секции **ELSE** осуществляется переход на оператор **END**.

При использовании конструкции с **GOTO** в случае истинного значения условного выражения <expression> выполняется переход на метку <label>. Первый символ метки должен быть буквенным, за которым следует некоторое количество алфавитно-цифровых символов или символ подчеркивания (_). Сразу же за меткой должен идти символ двоеточия (:).

Примеры

```
IF (EN) THEN  
  BEGIN  
    STATE Q0 = UNKNOWN  
  ELSE
```

```
IF (IN2) THEN
  BEGIN
    STATE Y2 = ONE
  ELSE
    STATE Y2 = ZERO
  END
END

IF (x = -2) THEN GOTO Do_Neg2
.
.
.
Do_Neg2:
.
.
.
```

Смотрите также

GOTO, WHILE ... DO

INPUTS

Объявляет входные выходы (выводы, управляющие работой схемы).

Основная форма

```
INPUTS <input pin>[, <input pin>,
```

Параметры

<input pin> Имя входного вывода.

Применение

Тип данных **INPUTS** используется для определения выводов, на которые поступают входные сигналы для данного устройства. Как правило, это входные и двунаправленные (I/O) выходы, выходы питания и заземления.

Замечания

Имена входных выводов обязательно должны начинаться с буквы и должны быть определены до их использования.

Пример

```
INPUTS VCC, GND, PRE, DATA, CLK, CLR
```

Смотрите также

OUTPUTS, IO_PAIRS, PWR_GND_PINS

INSTANCE

Проверяет определен ли экземпляр устройства.

Основная форма

```
INSTANCE("<instance name>")
```

Параметры

<instance name> Текстовая строка, содержащая имя экземпляра.

Применение

Эта функция возвращает 1, если присутствующий экземпляр SimCode устройства соответствует указанному имени <instance name>, иначе возвращается 0.

Замечания

Схема может содержать несколько устройств одного типа, а во процессе моделирования может понадобиться информация о том, интересующее ли устройство в данный момент моделируется. Например, с помощью этого оператора можно вывести на печать для дальнейшего просмотра сообщения для одного конкретного устройства И-НЕ и, таким образом, избавиться от необходимости разбирать большой набор сообщений от всех элементов И-НЕ в схеме. Имя экземпляра — это символическое обозначение устройства с символом его SPICE префикса (буква A).

Пример

```
IF (INSTANCE("AU23")) THEN
  BEGIN
    MESSAGE("U23-Q0 = %d", Q0);
  END;
```

INTEGERS

Объявляет целые переменные и массивы.

Основная форма

```
INTEGERS <var>[, <var>, ...];
```

Параметры

<var> Имя переменной.

Применение

Тип данных **INTEGERS** используется для определения целых переменных и массивов.

Замечания

Целые переменные и массивы должны начинаться с буквы и должны быть определены перед их первым использованием. Массивы определяются с помощью имени и квадратных скобок ([,]), между которыми располагается целое число, указывающее размер массива. Массивы целых чисел могут быть использованы в выражениях, а также с помощью выражений можно присваивать значения элементам массива.

В таблице 4.54 перечислены зарезервированные переменные SimCode целого типа, не требующие объявления.

Таблица 4.54

Переменная	Применение	Параметр цифровой модели	Опция SPICE
tp_param	tplh/hl индекс	Задержки распространения	TPMNTYMX
tt_param	ttlh/hl индекс	Длительности фронтов	TTMNTYMX
ld_param	LOAD индекс	Входная нагрузка	LDMNTYMX
drv_param	DRIVE индекс	Выходная нагрузка	DRVMNTYMX
i_param	ICC индекс	Ток устройства	IMNTYMX
user_param	USER индекс	Определенное пользователем	USERMNTYMX
warn_param	Предупреждение	Флаг WARN	SIMWARN
init_sim	1 во время инициализации SimCode	N/A	N/A
tran_pin	TRAN_xx индекс вывода	N/A	N/A

Первые шесть переменных в этой таблице могут принимать значения 1, 2 или 3. Эти значения являются индексами в массиве min/typ/max:

Значение	Описание
1	Индекс минимального значения.
2	Индекс типового значения.
3	Индекс максимального значения.

Параметр цифровой модели может быть установлен в диалоговом окне **Digital Model Parameters** независимо для каждого цифрового устройства. Если настройки параметров **SPICE Option** были сделаны в диалоговом окне **Analog Options**, то эти установки перекроют установки параметра цифровой модели для всех цифровых устройств. Явное присваивание значения какой-либо переменной в SimCode отменит все ранее сделанные настройки.

Переменная **warn_param** может быть установлена в любое положительное значение для вывода на экран при определенных условиях предупреждения для текущего устройства. С помощью ввода различных значений можно реализовать несколько уровней предупреждений. Значение **init_sim** устанавливается в 1 во время инициализации SimCode программы, иначе оно равно 0. Значение **tran_pin** равно индексу того вывода, который устанавливается в операторе **DELAY CASE**. Этот индекс используется для определения вывода, к которому применяется оператор **TRAN_xx**.

Пример

```
INTEGERS tblIndex, count, data[64];
```

Смотрите также

DELAY, MIN_TYP_MAX

IO_PAIRS

Объявляет связи между входными и выходными выводами для входной нагрузки.

Основная форма

```
IO_PAIRS (<ipin:opin>[, <ipin:opin>,
```

Параметры

<ipin:opin> Имена выводов из связанных входных и выходных выводов.

Применение

Оператор **IO_PAIRS** определяет связь выводов из списка **INPUTS** с выводами из списка **OUTPUTS**, что в дальнейшем используется оператором **LOAD**.

Замечания

Каждый физический входной вывод устройства имеет входную (**ipin**) и выходную (**opin**) секции SimCode кода. Секция **opin** требуется для обеспечения входных нагрузочных характеристик. Рассматриваемый оператор можно использовать в SimCode программе только один раз. Выводы питания в оператор **IO_PAIRS** не включаются.

Пример

```
IO_PAIRS (IN1:IN1_LD, IN2:IN2_LD
```

В данном примере выводы **IN1** и **IN2** являются входными (содержатся в списке **INPUTS**), а **IN1_LD** и **IN2_LD** — выходными (содержатся в списке **OUTPUTS**). Выводы **IN1** и **IN1_LD** ассоциируются с одним и тем же физическим выводом в реальном устройстве.

Смотрите также

INPUTS, OUTPUTS, LOAD

LEVEL

Устанавливает уровень для какого-либо состояния выхода.

Основная форма 1

```
LEVEL <output> [<output> ...] = (<expression
```

Основная форма 2

```
LEVEL <output> [<output> ...] = {ZERO}|{ONE}|{UNKNOWN
```

Параметры

<output> Имя или индексная переменная выхода.

<expression> Выражение, сравниваемое с **VOL** или **VON**.

Применение

Состояние выходного вывода определяется посредством его уровня и мощности. Команда **LEVEL** используется для установки уровня одного или более выходных выводов.

Таблица 4.55

<expression>	Состояние	Уровень
<= vol_param	ZERO	vol_param
>= voh_param	ONE	voh_param
другое	UNKNOWN	v3s_param

Замечания

Выходные выводы в команде могут быть заданы либо с помощью их имени, либо посредством переменной целого типа, которая содержит индекс данного вывода, но в одной команде не допускается применение обоих способов.

Примеры

```
LEVEL Q = ONE;
LEVEL Q1 Q2 Q3 Q4 = ZERO;
LEVEL OUT = ((1+2)/3);
```

В последнем примере OUT примет значение:

ZERO если $vil_param > 1$

UNKNOWN если $vil_param < 1$ и $vih_param > 1$

ONE если $vih_param < 1$

Смотрите также

REALS, STATE, STATE_BIT, STRENGTH

LOAD

Объявляет нагрузочные характеристики входных выводов.

Основная форма

```
LOAD <output> [<output> ...] =
    (v0=<value> r0=<value> [v1=<value> r1=<value>] [io=<value>] t=<value>);
```

Параметры

<output> Имя или индексная переменная выходного вывода.

<value> Вещественное значение или переменная.

v0 Напряжение нагрузки для высокого (**HIGH**) уровня.

r0 Сопротивление нагрузки для высокого (**HIGH**) уровня.

v1 Напряжение нагрузки для низкого (**LOW**) уровня.

<i>r1</i>	Сопротивление нагрузки для низкого (LOW) уровня
<i>io</i>	Сопротивление нагрузки в выключенном состоянии для неиспользуемой нагрузки.
<i>t</i>	Временная задержка до момента приложения нагрузки.

Применение

Команда **LOAD** обычно используется для входных и питающих выводов для задания нагрузки управляющей цепи. Так как нагрузку могут обеспечить только выходные выводы, каждый вход должен быть связан с соответствующим выходом, что выполняется с помощью оператора **IO_PAIRS**.

Если для разных входов требуются разные нагрузки, тогда можно применить команду **LOAD** несколько раз. Выводы питания необходимо размещать в отдельной команде **LOAD**, которая не включает параметры **v1/r1** и **io**. Выводы питания не входят в оператор **IO_PAIRS**. Оператор **IO_PAIRS** должен стоять в программе раньше любой из команд **LOAD**, содержащих параметр **io**.

Замечания

Входная нагрузка описывается из напряжения и сопротивления (**v0/r0** или **v1/r1**). Уровень напряжения входного сигнала будет определять, какая из этих пар будет использована. Если в начале уровень был ниже **VIL** и остается ниже **VIH**, то считается, что установлен низкий (**LOW**) уровень, и используются параметры **v1/r1**. Если в начале уровень был выше **VIH** и остается выше **VIL**, то считается, что установлен высокий (**HIGH**) уровень, и используются параметры **v0/r0**. Параметр **io** задает сопротивление нагрузки в выключенном состоянии. Неиспользуемую нагрузку можно удалить из схемы с помощью изменения значения ее сопротивления (параметра **r**) на значение, определенное для **io**.

Значения параметров **v0**, **r0**, **v1** и **r1** могут быть либо вещественными константами, либо вещественными переменными, а **io** и **t** должны быть только вещественными константами. Имена и индексные переменные выводов одновременно использовать в одной команде **LOAD** нельзя.

Для входных выводов значение **r0** определяется из справочных данных из параметра **III**. Например, вход стандартной микросхемы типа **LS** дает максимальный ток 20 мкА при напряжении **Vin** = 2.7 В. Таким образом, если **vol_param** = 0.2 В, то:

для максимальной нагрузки в высоком состоянии:

$$\begin{aligned} r0 &= (V_{in} - vol_param) / III_{max} \\ r0 &= (2.7V - 0.2V) / 20\mu A \\ r0 &= 125k\ ohms \end{aligned}$$

Значение **r1** извлекается из справочных данных из параметра **III**. Та же микросхема дает максимальный ток 400 мкА при напряжении **Vin** = 0.4 В. Таким образом, если **voH_param** = 4.6 В, то:

для максимальной нагрузки в низком состоянии:

$$\begin{aligned} r1 &= (voH_param - V_{in}) / III_{max} \\ r1 &= (4.6V - 0.4V) / 400\mu A \\ r1 &= 10.5k\ ohms \end{aligned}$$

Для выводов питания значение $r0$ извлекается из параметра **ICC**. Для микросхемы 74LS151 типовое значение тока питания равно $I_{cc} = 6$ мА при напряжении $V_{cc} = 5$ В и максимальный ток $I_{cc} = 10$ мА при напряжении $V_{cc} = 5.25$ В. Таким образом:

для типового значения I_{cc} : $r0 = 5V / 6mA = 833$ ohms

для максимального значения I_{cc} : $r0 = 5.25V / 10mA = 525$ ohms

При создании многоэлементного устройства, такого как микросхема 74LS00, содержащего четыре секции И-НЕ, необходимо устанавливать нагрузку по питанию индивидуально для каждого элемента.

Examples

```
r0_val = (MIN_TYP_MAX(ld_param: NULL, NULL, 125k);
r1_val = (MIN_TYP_MAX(ld_param: NULL, NULL, 10.5k);
ricc_val = (MIN_TYP_MAX(ld_param: NULL, 833, 525);
LOAD PRE_LD DATA_LD CLK_LD CLR_LD = (v0=v0l_param, r0=r0_val,
v1=voh_param, r1=r1_val, io=1e9, t=1p);
LOAD VCC_LD = (v0=gnd_param, r0=ricc_val, t=1p);
```

Смотрите также

IO_PAIRS, DRIVE

Математические функции

Таблица 4.56

Функция	Описание	Пример
POW	Степень	X= (12 POW(3));
ABS	Модуль	X= (ABS(-12));
SQRT	Квадратный корень	X= (SQRT(2));
EXP	Экспоненциальная функция	X= (EXP(10));
LOG	Натуральный логарифм	X= (LOG(0.1));
LOG10	Десятичный логарифм	X= (LOG10(0.1));
SIN	Синус	X= (SIN(0.1));
COS	Косинус	X= (COS(0.1));
TAN	Тангенс	X= (TAN(0.1));
ASIN	Арсинус	X= (ASIN(0.1));
ACOS	Аркуосинус	X= (ACOS(0.1));
ATAN	Арктангенс	X= (ATAN(0.1));
HSIN	Гиперболический синус	X= (HSIN(0.1));
HCOS	Гиперболический косинус	X= (HCOS(0.1));
HTAN	Гиперболический тангенс	X= (HTAN(0.1));

Смотрите также

Знаки операций

MESSAGE

Выводит сообщение без остановки моделирования.

Основная форма

```
MESSAGE("<message>"[, <value/pin>...]);
```

Параметры

- <message> Строка сообщения, включающая, если необходимо, символы форматирования.
- <value> Переменная или константа.
- <pin> Имя вывода или переменная с индексом вывода.

Применение

Оператор **MESSAGE** используется для вывода текстовой строки **<message>** без прерывания процесса моделирования. Выводимое сообщение появляется в статусном окне во время моделирования.

Замечания

Форматированная строка в операторе **MESSAGE** подобна той, которая используется в языке Си в операторе **printf**. Ниже в таблице 4.57 приведены некоторые допустимые символы форматирования:

Таблица 4.57

Символ	Описание
\t	Табуляция
\n	Новая строка
\r	Возврат каретки
%d	Вывод в десятичной форме для коротких переменных или текущее состояние входа/выхода
%D	Вывод в десятичной форме для коротких переменных или старое состояние входа/выхода
%x	Вывод в шестнадцатеричной форме для коротких переменных или текущее состояние входа/выхода
%X	Вывод в шестнадцатеричной форме для коротких переменных или старое состояние входа/выхода
%c	Символьный вывод для коротких переменных или текущее состояние входа/выхода
%C	Символьный вывод для коротких переменных или старое состояние входа/выхода
%e	Вывод в экспоненциальной форме вещественных переменных
%f	Вывод вещественных переменных в виде числа с плавающей точкой
%g	Вывод в короткой форме (%e или %f) вещественных переменных
%s	Вывод строковой константы

Замечание: в качестве строковых могут быть использованы только следующие константы:

INSTANCE Имя экземпляра SimCode устройства.
 FUNC Имя функции SimCode устройства.
 FILE Имя файла SimCode устройства.

Examples

```
MESSAGE("time\t\tCLK\tDATA\tQ\tQN");
MESSAGE("device instance= %s", INSTANCE);
MESSAGE("%.3e\t%d\t%d\t%d\t%d", present_time, CLK, DATA, Q, QN);
```

Смотрите также

PROMPT

MIN_TYP_MAX

Возвращает значение из поисковой таблицы **MIN_TYP_MAX**.

Основная форма

```
MIN_TYP_MAX(<index>: <min>, <typ>, <max>);
```

Параметры

- <index> Входная переменная (индекс минимального, типичного или максимального значения).
- <min> Минимальное значение из справочника.
- <typ> Типичное значение из справочника.
- <max> Максимальное значение из справочника.

Применение

Функция **MIN_TYP_MAX** похожа на функцию **SELECT_VALUE** за исключением того, что здесь нужно вводить три значения переменных. Ключевое слово **NULL** может являться заменой для одного или двух неизвестных значений. Если предопределенная целая переменная (смотри **INTEGER**) используется в качестве индекса (<index>), неизвестные (**NULL**) значения вычисляются из известных по формулам, приведенным в таблице 4.58.

Таблица 4.58

Известные значения	Формула
<min>, <max>	$typical = (<max> + <min>) / 2;$
только <min>	$typical = (<min> / <min\ scale\ factor>)$
	$maximum = (<min> / <min\ scale\ factor>) * <max\ scale\ factor>$
только <typ>	$minimum = (<typ> * <min\ scale\ factor>)$
	$maximum = (<typ> * <max\ scale\ factor>)$
только <max>	$minimum = (<max> / <max\ scale\ factor>) * <min\ scale\ factor>$
	$typical = (<max> / <max\ scale\ factor>)$

Замечания

Если параметр <index> не является одной из предопределенных переменных, перечисленных ниже, тогда <мин. масштабный коэффициент>=0.5, а <макс. масштабный коэффициент>=1.5. Значения этих коэффициентов для любой предопределенной переменной может быть изменено в диалоговом окне Analog Options. Значения этих коэффициентов для **ld_param**, **drv_param** и **i_param** реверсируются, так как эти параметры устанавливают значения сопротивлений, а не токов (т.е. максимальная нагрузка соответствует минимальному сопротивлению).

Таблица 4.59

Переменная	SPICE опция	Параметр	По умолчанию
tp_param	PROPMNS	<min scale factor>	0.5
tp_param	PROPMXS	<max scale factor>	1.5
tt_param	TRANMNS	<min scale factor>	0.5
tt_param	TRANMXS	<max scale factor>	1.5
ld_param	LOADMNS	<min scale factor>	1.5
ld_param	LOADMXS	<max scale factor>	0.5
drv_param	DRIVEMNS	<min scale factor>	1.5
drv_param	DRIVEMXS	<max scale factor>	0.5
i_param	CURRENTMNS	<min scale factor>	1.5
i_param	CURRENTMXS	<max scale factor>	0.5
vth_param	VTHMNS	<min scale factor>	0.5
vth_param	VTHMXS	<max scale factor>	1.5
user_param	USERMNS	<min scale factor>	0.5
user_param	USERMXS	<max scale factor>	1.5

Примеры

```
tplh_val = (MIN_TYP_MAX(tp_param: NULL, 5n, NULL));
```

В этом примере если принять, что **PROPMNS** и **PROPMXS** установлены по умолчанию, то:

если *tp_param* = 1, то *tplh_val* = 2.5n

если *tp_param* = 2, то *tplh_val* = 5n

если *tp_param* = 3, то *tplh_val* = 10n

```
ricch_val = (MIN_TYP_MAX(i_param: NULL, 2500, 1250));
```

В этом примере если принять, что **CURRENTMNS** и **CURRENTMXS** установлены по умолчанию, то:

если *i_param* = 1, то *ricch_val* = 5000

если *i_param* = 2, то *ricch_val* = 2500

если *i_param* = 3, то *ricch_val* = 1250

Смотрите также

INTEGERS, SELECT_VALUE

NO_CHANGE

Оставляет неизменным выходное состояние I/O вывода.

Основная форма

```
NO_CHANGE <output> [<output> ...];
```

Параметры

<output> Имя или индексная переменная выходного вывода.

Применение

Функция **NO_CHANGE** используется для установки флага неизменяемого состояния для указанных выходных выводов. Эта функция также применяется к двунаправленным выводам, когда они используются в качестве входов.

Замечания

Функция **NO_CHANGE** используется для установки флага неизменяемого состояния для указанных выходных выводов. Эта функция также применяется к двунаправленным выводам, когда они используются в качестве входов.

Пример

```
NO_CHANGE Q1 Q2 Q3 Q4;
```

NUMBER

Возвращает число, полученное на основе состояний выводов, которым присвоены двоичные весовые коэффициенты.

Основная форма

```
NUMBER(<MSB pin>, [<pin>, ...] <LSB pin> );
```

Параметры

<pin> Имя или индекс вывода.

Применение

Функция **NUMBER** возвращает короткое целое (**short integer**) число, равное десятичному значению двоичного числа, представленного параметрами **<pin>**. Каждый бит, представленный параметром **<pin>**, принимается либо за 1, если **<pin>** не равен нулю, либо за 0.

Замечания

Первый параметр **<pin>** в списке (**<MSB pin>**) является старшим битом, а последний — младшим (**<LSB pin>**).

Пример

```
A = (NUMBER (D3, D2, D1, D0));
```

В этом примере, если D3 имеет высокий уровень (**HIGH**), а D2, D1 и D0 — низкие (**LOW**), то $A = 8_{10}$ (1000_2).

Знаки операций

Операция присваивания

= Равенство (устанавливает переменную или выходной вывод в какое-либо значение или состояние).

Математические операции

+	Сложение
-	Вычитание
*	Умножение
/	Деление

Унарные операции

~	Логическое отрицание (НЕ)
!	Поразрядное дополнение

Логические операции

&&	И (AND)
	ИЛИ (OR)
^^	исключающее ИЛИ (XOR)

Поразрядные операции

&	И (AND)
	ИЛИ (OR)
^	исключающее ИЛИ (XOR)
<<	сдвиг влево
>>	сдвиг вправо

Операции сравнения

=	равно
!=	не равно
<	меньше
<=	меньше или равно
>	больше
>=	больше или равно

Применение

Знаки операций используются для установки значений и манипулирования переменными и выражениями.

Замечания

Выражения должны быть заключены в круглые скобки. Выражение внутри скобок всегда вычисляются слева на право. Для изменения порядка вычисления операций в выражении необходимо также использовать скобки. Значения, переменные, выражения и т. п., к которым применяются унарные операции, должны также заключаться в скобки.

Примеры

```
clk_twl = (25n);
```

```

reg = (reg + 1);
vx = (vol_param - 10m);
C = (A * B);
val = (xval / 2);
X = (A && ~(B));
Y = (!X);           //if X=1 then Y=FFFFFFFE
A = (X & 1);        //if X=1 then A=1, if X=2 then A=0
B = (X | 8);        //if X=1 then B=9, if X=2 then B=10
C = (X >> 2);       //if X=1 then C=0, if X=2 then C=0
D = (2 >> X);       //if X=1 then D=1, if X=2 then D=0
E = (X << 2);       //if X=1 then E=4, if X=2 then E=8
F = (2 << X);       //if X=1 then F=4, if X=2 then F=8
IF (A >= B) THEN ...
IF ((A < 2) && (B > 3)) THEN ...
IF ((C < 2) || (X > 4)) THEN ...

```

Смотрите также

MATH FUNCTIONS

OUTPUTS

Объявляет выходные выводы (выводы, являющиеся управляющими или нагружающими для внешней схемы).

Основная форма

```
OUTPUTS <output pin>[, <output pin>, ...];
```

Параметры

<output pin> Имя выходного вывода.

Применение

Тип данных **OUTPUTS** используется для определения выводов, которые во время работы устройства воздействуют на внешнюю схему. В основном они включают входные, выходные и I/O выводы, а также выводы питания. Входные выводы и выводы питания включаются в этот список по причине того, что их наличие позволяет задать нагрузку для внешних управляющих цепей.

Замечания

Имена выходных выводов обязательно должны начинаться с буквы и быть определены до их первого использования.

Пример

```
OUTPUTS VCC_LD, PRE_LD, DATA_LD, CLK_LD, CLR_LD, QN, Q;
```

Смотрите также

INPUTS, IO_PAIRS, PWR_GND_PINS

PARAM_SET

Определяет, была ли выполнена предварительная установка SimCode параметра.

Основная форма

```
PARAM_SET(<param var>)
```

Параметры

<param var> Параметр SimCode модели.

Применение

Функция **PARAM_SET** используется для определения того, было ли установлено какое-либо значение для указанного параметра SimCode модели. Она возвращает 1, если значение было установлено (например, `vil_param = 0.8`), в противном случае возвращается 0.

Замечания

Список параметров SimCode модели и ассоциированные с ними имена переменных приведены в описании команд **INTEGER** и **REAL**.

Пример

```
A = PARAM_SET(ld_param);  
IF (PARAM_SET(voh_param)) THEN ...
```

Смотрите также

INTEGERS, REALS

PROMPT

Приостанавливает процесс моделирования и выводит сообщение.

Основная форма

```
PROMPT("<message>" [, <value/pin>...]);
```

Параметры

<message> Строка сообщения, содержащая, если необходимо, символы форматирования.

<value> Значение переменной или константы.

<pin> Имя вывода или переменная с индексом вывода.

Применение

Оператор **PROMPT** используется для остановки процесса моделирования и вывода информации из строки **<message>**. Сообщение во время моделирования выводится в окне статуса. Для продолжения выполнения SimCode кода необходимо нажать на соответствующую кнопку.

Замечания

Форматированная строка в операторе **PROMPT** подобна той, которая используется в языке Си в операторе **printf**. В таблице приведены некоторые допустимые символы форматирования:

Таблица 4.60

Символ	Описание
\t	Табуляция
\n	Новая строка
\r	Возврат каретки
%d	Вывод в десятичной форме для коротких переменных или текущее состояние входа/выхода
%D	Вывод в десятичной форме для коротких переменных или старое состояние входа/выхода
%x	Вывод в шестнадцатеричной форме для коротких переменных или текущее состояние входа/выхода
%X	Вывод в шестнадцатеричной форме для коротких переменных или старое состояние входа/выхода
%c	Символьный вывод для коротких переменных или текущее состояние входа/выхода
%C	Символьный вывод для коротких переменных или старое состояние входа/выхода
%e	Вывод в экспоненциальной форме вещественных переменных
%f	Вывод вещественных переменных в виде числа с плавающей точкой
%g	Вывод в короткой форме (%e или %f) вещественных переменных
%s	Вывод строковой константы

Замечание: в качестве строковых могут быть использованы только следующие константы:

INSTANCE Имя экземпляра SimCode устройства.
FUNC Имя функции SimCode устройства.
FILE Имя файла SimCode устройства.

Пример

```
PROMPT("input=%d time=%f device=%s", D1, t1, INSTANCE);
```

Смотрите также

MESSAGE

PWL_TABLE

Возвращает значение из интерполяционной поисковой таблицы.

Основная форма

```
PWL_TABLE (<IN var>: <IN1>, <OUT1>, <IN2>, <OUT2> [, ... <OUTn>, <OUTn>])
```

Параметры

<IN var> Входная переменная (целая или вещественная)

<INx> Входное сравниваемое значение
<OUTx> Выходное значение для <INx>

Применение

Эта кусочно-линейная функция по существу является поисковой таблицей. Значение параметра <IN var> используется для поиска входа в таблицу, состоящую из пар значений. Первое значение в каждой паре — входное сравниваемое значение, второе — соответствующее первому выходное значение. Если <IN var> меньше первого <IN> в списке, то возвращается значение первого <OUT>. Если <IN var> больше последнего <INn> в списке, то возвращается значение последнего <OUTn>. Между входными значениями выполняется линейная интерполяция согласно следующей формулы:

$$\text{value} = (((\text{OUTA} - \text{OUTB}) / (\text{INA} - \text{INB})) * (\text{IN var} - \text{INA}) + \text{OUTA})$$

где <IN var> лежит между входными значениями **INA** и **INB**. Вычисленное выходное значение будет лежать между **OUTA** и **OUTB**.

Замечания

В параметрах описываемой функции должно содержаться две или более пар входных/выходных значений, а входные значения должны быть перечислены в возрастающем порядке. На максимальное количество пар ограничение не накладывается.

Пример

```
twh = (PWL_TABLE(var: 5,180n,10,120n,15,80n));
```

В этом примере, если var = 10, то twl = 120 нс, а если var = 12, то twl = 104.

Смотрите также

SELECT_VALUE

PWR_GND_PINS

Объявляет выводы питания и заземления и устанавливает значения напряжений на них.

Основная форма

```
PWR_GND_PINS (<pwrpin>, <gndpin>);
```

Параметры

<pwrpin> Имя вывода питания.
<gndpin> Имя вывода заземления.

Применение

Оператор **PWR_GND_PINS** определяет какие из входных выводов (перечисленных в команде **INPUTS**) являются выводами питания и заземления, а также устанавливает абсолютные величины напряжений для параметров **Power** и **Ground** устройства следующим образом:

pwr_param = напряжение на <pwrpin>
gnd_param = напряжение на <gndpin>

Замечания

Этот оператор может быть использован в SimCode только один раз. Только один вывод может быть определен в качестве вывода питания и один как вывод заземления.

Пример

```
PWR_GND_PINS (VCC, GND);
```

Смотрите также

INPUTS, OUTPUTS, REALS, VIL_VIH_PERCENT, SUPPLY_MIN_MAX

READ_DATA

Читает данные из ASCII файла в массивы.

Основная форма

```
READ_DATA(<array>[, <array>, ...])
```

Параметры

<array> Имя массива, заполняемого данными из файла.

Применение

Оператор **READ_DATA** открывает файл, имя которого задано параметром **data=** в операторе **.MODEL**, и читает данные из этого файла, представленные в ASCII формате. Количество и тип (целый или вещественный) считываемых значений, расположенных в каждой строке, определяется по количеству и типу переменных в массивах, указанных в операторе. Количество прочитанных данных ограничено либо числом строк в файле, либо размером наименьшего массива. Возвращаемое функцией **READ_DATA** значение равно количеству прочитанных строк. В случае возникновения ошибки возвращается отрицательное число:

- 1 Неправильное имя файла
- 2 Невозможно найти файл
- 3 Некорректно заданный массив
- 4 Неверный доступ к массиву
- 5 Неверный тип данных
- 6 Ожидалось значение данных

Замечания

Если в каждой строке файла расположено несколько значений, они должны быть разделены запятой. Данные вещественного типа в файле должны быть представлены в экспоненциальном виде. Оператор **.MODEL**, содержащий параметр **data=**, должен быть помещен в файл **MDL** данного устройства.

Пример

Файл **MYDEVICE.MDL**:

```
.MODEL AMYDEVICE XSIMCODE (file="{MODEL_PATH}MYDEVICES.SCB" + func=MyDevice  
data="{MODEL_PATH}MYDEVICE.DAT" {mntymx})
```

Файл **MYDEVICE.DAT**:

```
8, 8E-6
```

9, 9E-6
 10, 1E-5
 11, 1.1E-5

SimCode код устройства **MyDevice**:

```
nlines = READ_DATA(int_array, real_array);
```

В этом примере открывается файл **MYDEVICE.DAT**, расположенный в каталоге моделей (**MODEL_PATH**). Затем читаются два столбца данных, содержащих соответственно целые и вещественные значения. Если массивы объявлены как **int_array[3]** и **real_array[3]**, тогда будут прочитаны только первые три строки и переменная **nlines** получит значение 3.

REALS

Объявляет переменные и массивы вещественного типа.

Основная форма

```
REALS <var>[, <var>, ...];
```

Параметры

<var> Имя переменной

Применение

Тип данных **REALS** используется для определения вещественных переменных и массивов.

Замечания

Вещественные переменные и массивы должны начинаться с буквы и быть определены до их первого использования. Массив с элементами вещественного типа определяется с помощью его имени и следующими за ним квадратными скобками, в которых указывается его размер. Вещественные массивы могут быть установлены и/или использованы в выражениях.

Далее в таблице 4.61 приведены зарезервированные SimCode вещественные переменные, которые не требуют объявления.

Таблица 4.61

Переменная	Использование	Параметр цифровой модели
vil_param	Низкий входной уровень	Значение VIL
vih_param	Высокий входной уровень	Значение VIH
vol_param	Низкий выходной уровень	Значение VOL
voh_param	Высокий выходной уровень	Значение VOH
v3s_param	Уровень выхода в Z-состоянии	N/A
rol_param	Значение мощности выхода при низком уровне	N/A

roh_param	Значение мощности выхода при высоком уровне	N/A
r3s_param	Значение мощности выхода в Z-состоянии	N/A
pwr_param	Напряжение вывода питания	Значение PWR
gnd_param	Напряжение вывода заземления	Значение GND
present_time	Настоящее время моделирования	N/A
previous_time	Предыдущее время моделирования	N/A
sim_temp	Рабочая температура схемы	N/A (Опция SPICE: TEMP)

Параметр цифровой модели для каждого цифрового устройства может быть установлен независимо в диалоговом окне **Digital Model Parameters**. Если переменная устанавливается непосредственно в SimCode коде, то это значение перекрывает все другие настройки.

Значения **pwr_param** и **gnd_param** устанавливаются каждый раз при выполнении оператора **PWR_GND_PINS**. Значения **present_time** и **previous_time** устанавливаются всякий раз, когда происходит переход на очередной временной шаг. Значение **sim_temp** представляют собой текущую рабочую температуру схемы, которая может быть установлена с помощью **SPICE** опции **TEMP**.

Пример

```
REALS tplh_val, tphl_val, ricc_val, vbias, values[64];
```

Смотрите также

PWR_GND_PIN, VIL_VIH_VALUE, VIL_VIH_PERCENT, VOL_VOH_MIN

RECOVER

Проверяет входы на наличие нарушений времени восстановления.

Основная форма

```
RECOVER(<clk input> = {LH}|{HL} <mr input> [<mr input> ...]
```

Параметры

<clk input> Имя или индекс входного тактового/опорного вывода, который подвергается тестированию.

<mr input> Имя или индекс входного вывода установки/сброса, который подвергается тестированию.

TREC Время восстановления и для высокого и для низкого уровня на выводе <mr pin>.

TRECL Время восстановления низкого уровня на выводе <mr pin>.

TRECH Время восстановления высокого уровня на выводе <mr pin>.

<message> Текстовая строка, которая будет выведена в качестве предупреждения.

Применение

Функция **RECOVER** сравнивает разницу во времени между изменением уровня (из 0 в 1 или из 1 в 0) на выводе **<clk input>** и изменением уровня на **<mr input>** с указанным тестовым временем. Значения этих временных промежутков определяются либо совместно в параметре **TREC=<time>** (то есть, **TRECL** и **TRECH** получают одинаковые значения), либо индивидуально с помощью параметров **TRECL=<time>** и **TRECH=<time>**. Если сравниваемое время меньше значения **<time>**, то в процессе моделирования появится соответствующее сообщение. Если в вызове функции присутствует необязательный параметр **<message>**, то его содержимое будет выведено в качестве этого сообщения.

Замечания

С описываемой функцией используются данные из справочника. Параметры **TRECL=<time>** и **TRECH=<time>** могут быть введены в одной функции **RECOVER**. Проверка времени восстановления может быть производиться только в случае, если состояние **<mr input>** соответствует указанному в параметре (низкий уровень для **TRECL** и высокий — для **TRECH**), когда на **<clk input>** происходит заданный переход (из 0 в 1 — **LH** или из 1 в 0 — **HL**). Например, если **<clk input> = LH** и указано значение **TRECL**, то для выполнения тестирования **<mr input>** должен быть в низком состоянии, когда **<clk input>** переходит от низкого к высокому. В одном операторе **RECOVER** могут использоваться одновременно и имена выводов и индексные переменные.

Пример

```
RECOVER(CLK=LH PRE CLR TREC=trec_val  
"CLK->PRE or CLR");
```

RETURN

Осуществляет выход из SimCode подпрограммы.

Основная форма

```
RETURN;
```

Применение

Оператор **RETURN** используется для возврата на инструкцию, следующую непосредственно за последним выполненным оператором **GOSUB**.

Смотрите также

GOSUB

SELECT_VALUE

Возвращает значение из простой поисковой таблицы.

Основная форма

```
SELECT_VALUE (<index>: <val/pin/var>,  
<val/pin/var>[, <val/pin/var>, ...]);
```

Параметры

<index> Индекс в списке параметров **<val/pin/var>**.

<val/pin/var> Выходное значение, вывод или переменная.

Применение

Функция **SELECT_VALUE** возвращает значение числа или переменной, которая определяется значением индексной переменной.

Замечания

Количество используемых значений и/или переменных не ограничено.

Пример

```
A = (SELECT_VALUE(B: 16, 8, 4, 2, 1));
```

В этом примере если $B = 2$, то $A = 8$ (второе значение).

Смотрите также

PWL_TABLE, MIN_TYP_MAX

SETUP_HOLD

Проверяет входной сигнал на предмет несоответствия времени установления (**SETUP**) и времени удержания (**HOLD**) заданными значениями.

Основная форма

```
SETUP_HOLD(<clk input> = {LH}|{HL}  
<data input> [<data input> ...]  
{TS=<time>}|{TSL=<time> TSH=<time>}  
{TH=<time>}|{THL=<time> THH=<time>} ["<message>"];
```

Параметры

<clk input> Имя или индекс входного тактового/опорного вывода, который подвергается тестированию.

<data input> Имя или индекс входного вывода данных, который подвергается тестированию.

TS Время установления для высокого и для низкого уровня на выводе <data input>.

TSL Время установления для низкого уровня на выводе <data input>.

TSH Время установления для высокого уровня на выводе <data input>.

TH Время удержания для высокого и для низкого уровня на выводе <data input>.

THL Время удержания для низкого уровня на выводе <data input>.

THH Время удержания для высокого уровня на выводе <data input>.

<message> Текстовая строка, которая будет выведена в качестве предупреждения.

Применение

Функция **SETUP_HOLD** сравнивает разницу во времени между изменением уровня (из 0 в 1 или из 1 в 0) на выводе <clk input> и изменением уровня на <data input> с указанным тестовым временем. Значения времени установления (**SETUP**) определяются ли-

бо совместно в параметре **TS=<time>** (**TSL** и **TSH** получают одинаковые значения), либо индивидуально с помощью **TSL=<time>** и **TSH=<time>**. Значения времен удержания (**HOLD**) определяются либо совместно в параметре **TH=<time>** (**THL** и **TNH** получают одинаковые значения), либо индивидуально с помощью **THL=<time>** и **TNH=<time>**. Если сравниваемое время меньше значения **<time>**, то появится соответствующее сообщение. Если в вызове функции присутствует необязательный параметр **<message>**, то его содержимое будет выведено в качестве такого сообщения.

Замечания

С описываемой функцией используются данные из справочника. **TSL=<time>**, **TSH=<time>**, **THL=<time>** и **TNH=<time>** могут быть введены в пределах одного оператора **SETUP_HOLD**. Тестирование времени установления и/или времени удержания может быть произведено только в случае, если состояние **<data input>** соответствует указанному в параметре (низкий уровень для **TSL** и **THL**, высокий — для **TSH** и **TNH**), когда на **<clk input>** происходит заданный переход (из 0 в 1 — **LH** или из 1 в 0 — **HL**). Например, если **<clk input> = LH** и указано значение **TSL**, то для выполнения тестирования времени установления **<data input>** должен быть в низком состоянии, когда **<clk input>** переходит от низкого к высокому. В одном операторе **SETUP_HOLD** могут использоваться одновременно и имена выводов и индексные переменные.

Пример

```
SETUP_HOLD(CLK=LH DATA Ts=ts_val Th=th_val "CLK->DATA");
```

STATE

Устанавливает выходы в объявленное логическое состояние.

Основная форма 1

```
STATE <output> [<output>...] = (<expression>);
```

Основная форма 2

```
STATE <output> [<output>...] = {ZERO}|{ONE}|{UNKNOWN};
```

Параметры

<output> Имя или индексная переменная выходного вывода.

<expression> Выражение, которое будет сравнено с **VIL** или **VIH**.

Применение

Состояние выходного вывода определяется его уровнем и мощностью. Оператор **STATE** устанавливает уровень и мощность для одного или более выходных выводов или переменных. Если **<expression>** меньше или равно **vil_param**, то выход будет установлен в ноль (**ZERO**). Если **<expression>** больше или равно **vih_param**, то выход будет установлен в единицу (**ONE**). Иначе выход получит неопределенное состояние (**UNKNOWN**). Значения уровня и мощности определяются следующим образом:

Таблица 4.62

<expression>	Состояние	Уровень	Мощность
<= vil_param	ZERO	vil_param	rol_param
>= vih_param	ONE	vih_param	roh_param

другое	UNKNOWN	v3s_param	r3s_param
--------	---------	-----------	-----------

Замечания

Выходные выводы могут быть определены либо с помощью своих имен, либо посредством переменной целого типа, которая содержит индекс выходного вывода. Имена и переменные одновременно не могут находиться в одной команде **STATE**.

Примеры

```
STATE Q = ONE;
STATE Q1 Q2 Q3 Q4 = ZERO;
STATE OUT = ((1+2)/3);
```

В последнем примере OUT примет значение:

ZERO если $vil_param > 1$

UNKNOWN если $vil_param < 1$ and $vih_param > 1$

ONE если $vih_param < 1$

Смотрите также

REALS, STATE_BIT, LEVEL, STRENGTH, TABLE, EXT_TABLE

STATE_BIT

Устанавливает логические состояния выходов согласно двоичным весовым коэффициентам.

Основная форма

STATE_BIT <output> [<output> ...] = (<expression>);

Параметры

<output> Имя или индексная переменная выходного вывода.

<expression> Выражение, которое может быть побитно соотносено с выводами.

Применение

Состояние выходного вывода определяется его уровнем и мощностью. Оператор **STATE_BIT** используется для установления уровня и мощности одного или более выходных выводов на основе значения параметра **<expression>**. Состояние первого вывода в списке устанавливается согласно первому (младшему) биту значения выражения, состояние второго вывода — согласно второму биту и т. д. Значения уровня и мощности определяются по принципу представленному в таблице 4.62.

Таблица 4.63

Значение бита	Состояние	Уровень	Мощность
0	ZERO	vol_param	rol_param
1	ONE	voh_param	roh_param

Замечания

Выходные выводы могут быть определены либо с помощью своих имен, либо посредством переменной целого типа, которая содержит индекс выходного вывода. Имена и переменные одновременно не могут находиться в одной команде **STATE_BIT**. Максимальное количество выходных выводов или переменных ограничено числом 16.

Пример

```
STATE_BIT Q1 Q2 Q3 Q4 = (internal_reg);
```

В этом примере если **internal_reg = 11** (1011 в двоичном виде), то **Q1** (младший значащий бит) = **ONE**, **Q2 = ONE**, **Q3 = ZERO** и **Q4** (старший значащий бит) = **ONE**.

Смотрите также

REALS, STATE, LEVEL, STRENGTH, TABLE, EXT_TABLE

STEP_OFF

Выключение режима трассировки SimCode кода.

Основная форма

```
STEP_OFF
```

Применение

Оператор **STEP_OFF** выключает режим трассировки (**TRACE**) SimCode кода.

Смотрите также

STEP_ON

STEP_ON

Включение режима трассировки SimCode кода.

Основная форма

```
STEP_ON
```

Применение

Оператор **STEP_ON** включает режим трассировки (**TRACE**) SimCode кода. Это приводит к тому, что перед выполнением каждой SimCode инструкции на экран выводится содержимое счетчика команд и сама эта инструкция.

Смотрите также

STEP_OFF

STRENGTH

Устанавливает мощность выхода

Основная форма 1

```
STRENGTH <output> [<output> ...] = (<expression>);
```

Основная форма 2

```
STRENGTH <output> [<output> ...] = {STRONG}|{HI_IMPEDANCE};
```

Параметры

<output> Имя или индексная переменная выходного вывода.

<expression> Выражение, которое будет использовано непосредственно как мощность.

Применение

Состояние выходного вывода определяется его уровнем и мощностью. Оператор STRENGTH используется для установки мощности одного или более выходных выводов.

Таблица 4.64

Значение	Состояние	Мощность
STRONG	ZERO	rol_param
STRONG	ONE	roh_param
HI_IMPEDANCE	N/A	r3s_param
<expression>	N/A	<expression>

Замечания

Выходные выводы могут быть определены либо с помощью своих имен, либо посредством переменной целого типа, которая содержит индекс выходного вывода. Имена и переменные одновременно не могут находиться в одной команде STRENGTH.

Смотрите также

REALS, STATE, STATE_BIT, LEVEL

SUPPLY_MIN_MAX

Проверяет выводы питания на предмет выхода напряжения за границы установленного диапазона.

Основная форма

```
SUPPLY_MIN_MAX(<min value>, <max value>);
```

Параметры

<min value> Минимальное рекомендованное напряжение питания.

<max value> Максимальное рекомендованное напряжение питания.

Применение

Функция **SUPPLY_MIN_MAX** проверяет разность напряжений между выводами питания и заземления. Эти выводы определены в операторе **PWR_GND_PINS**. Если в диалоговом окне **Digital Model Parameters** установлен флаг **WARN flag** и разность напряжений (**pwr_param – gnd_param**) меньше **<min value>** или больше **<max value>**, то в процессе моделирования будет выведено соответствующее предупреждение.

Замечания

С описываемой функцией используются данные из справочника. Для использования данной функции необходимо вызвать оператор **PWR_GND_PINS**.

Пример

```
SUPPLY_MIN_MAX(4.75, 5.25);
```

Смотрите также

INTEGERS, PWR_GND_PINS

TABLE

Устанавливает логические состояния выводов на основе таблицы истинности.

Основная форма

```
TABLE <line>
<input> [<input> ...] <output pin> [<output pin> ...]
<input state> [<input state> ...] <output state> [<output state> ...];
```

Параметры

<line>	Переменная, хранящая используемую строку в таблице.
<input>	Имя или индексная переменная входного вывода.
<output pin>	Имя выходного вывода.
<input state>	Состояние отдельных входов.
<output state>	Состояние отдельных выходов на основе входных условий.

Применение

Оператор **TABLE** действует как таблица истинности и используется для установления уровня и мощности указанных выходов. Допустимые состояния входов:

0	низкий (напряжение на входе $\leq v_{il_param}$).
1	высокий (напряжение на входе $\geq v_{ih_param}$).
X	напряжение на входе не важно.

Допустимые состояния выходов:

L	ZERO (уровень выхода = vol_param)
H	ONE (уровень выхода = voh_param).
Z	UNKNOWN (уровень выхода = $v3s_param$).

За символом состояния выхода может следовать двоеточие и символ, обозначающий мощность:

s	STRONG (установка выхода в rol_param при L и roh_param при H).
z	HI_IMPEDANCE (установка выхода в $r3s_param$).

Если символ мощности не указан, тогда будет использован **STRONG** при низком (L) и высоком (H) состоянии и **HI_IMPEDANCE** при состоянии Z.

Замечания

Каждая строка проверяется последовательно от верхней до нижней до тех пор, пока не встретятся входные условия. Выходы устанавливаются в соответствии с первой строкой, где встретились входные условия. В параметре **<line>** будет сохранен номер использованной строки. Если соответствия входным условиям не было обнаружено, то **<line>** примет значение 0. Использование имен и переменных в одной команде TABLE одновременно не допускается.

Пример

```
TABLE tblIndex
INA          INB   OUT
0            0     H
0            1     H
1            0     H
1            1     L;
```

В этом примере представлен двухвходовой элемент И-НЕ. Если оба входных вывода (**INA** и **INB**) находятся в высоком состоянии (**>=vih_param**), **OUT** устанавливается в ноль (**ZERO**) (**vol_param**) и **STRONG** (**rol_param**), а **tblIndex** будет равен 4.

Смотрите также

REALS, STATE, STATE_BIT, EXT_TABLE

VALUE

Возвращает значение указанного вывода.

Основная форма

```
VALUE(<pin>)
```

Параметры

<pin> Имя или индексная переменная вывода.

Применение

Функция **VALUE** возвращает число вещественного типа, показывающее уровень напряжения на указанном в параметре **<pin>** выводе.

Пример

```
v = (VALUE(D3));
```

VIL_VIH_PERCENT

Устанавливает значения **VIL** и **VIH** в процентах от напряжения питания.

Основная форма

```
VIL_VIH_PERCENT (<vil %>, <vih %>);
```

Параметры

<vil %> Процент от напряжения питания, определяющий **vil**.

<vih %> Процент от напряжения питания, определяющий **vih**.

Применение

VIL и **VIH** не используют массив мин/тип/макс для выбора своих значений, а должны быть явно объявлены для каждого цифрового устройства. Оператор **VIL_VIH_PERCENT** устанавливает **VIL** и **VIH** параметры устройства в процентном отношении к напряжению питания следующим образом:

$$\begin{aligned}vil_param &= (pwr_param - gnd_param) * \langle vil \% \rangle \\vih_param &= (pwr_param - gnd_param) * \langle vih \% \rangle\end{aligned}$$

Замечания

Для использования описываемой функции необходимо определить выводы питания в операторе **PWR_GND_PINS**. Значение процентов должно лежать в диапазоне от 0 до 100. Значения **vil_param** и **vih_param**, установленные с помощью оператора **VIL_VIH_PERCENT**, перекрывают значения, определенные в диалоговом окне **Digital Model Parameters** в полях **VIL value** и **VIH value**.

Пример

```
VIL_VIH_PERCENT(33, 67);
```

Смотрите также

REALS, **PWR_GND_PINS**, **VIL_VIH_VALUE**

VIL_VIH_VALUE

Устанавливает абсолютные значения параметров **VIL** и **VIH**.

Основная форма

```
VIL_VIH_VALUE (<vil>, <vih>);
```

Параметры

<vil> Абсолютный уровень напряжения, определяющий **vil**.
<vih> Абсолютный уровень напряжения, определяющий **vih**.

Применение

VIL и **VIH** не используют массив мин/тип/макс для выбора своих значений, а должны быть явно объявлены для каждого цифрового устройства. Оператор **VIL_VIH_VALUE** устанавливает **VIL** и **VIH** параметры устройства в абсолютные значения напряжений следующим образом:

$$\begin{aligned}vil_param &= \langle vil \rangle \\vih_param &= \langle vih \rangle\end{aligned}$$

Замечания

Для построения более точной модели действительных характеристик переключения цифрового входа значения **VIL** и **VIH** в общем случае не рекомендуется устанавливать значения из справочника. Исключения составляют случаи устройств с определенным гистерезисом, такие как 74LS14. Обычно гистерезис цифрового устройства мал и составляет примерно 100 мВ, но он никогда не равен 0 В.

Значения **vil_param** и **vih_param**, установленные с помощью оператора **VIL_VIH_VALUE**, перекрывают значения, определенные в диалоговом окне **Digital Model Parameters** в полях **VIL value** и **VIH value**.

Пример

```
VIL_VIH_VALUE(1.25, 1.35);
```

Смотрите также

REALS, VIL_VIH_PERCENT

VOL_VOH_MIN

Устанавливает значения **VOL** и **VOH** в зависимости от питания и заземления.

Основная форма

```
VOL_VOH_MIN (<vol offset>, <voh offset>, <min voh-vol>);
```

Параметры

- | | |
|---------------|--|
| <vol offset> | Напряжение смещения, которое должно быть приложено к напряжению вывода заземления для получения vol . |
| <voh offset> | Напряжение смещения, которое должно быть приложено к напряжению вывода питания для получения voh . |
| <min voh-vol> | Минимально допустимая разница между значениями vol и voh . |

Применение

VOL и **VOH** не используют массив мин/тип/макс для выбора своих значений, а должны быть явно объявлены для каждого цифрового устройства. Оператор **VOL_VOH_MIN** устанавливает **VOL** и **VOH** параметры устройства следующим образом:

$$vol_param = gnd_param + \langle vol\ offset \rangle$$
$$voh_param = pwr_param + \langle voh\ offset \rangle$$

Замечания

Для построения более точной модели действительных характеристик переключения цифрового входа значение **VOH** в общем случае не рекомендуется устанавливать значения из справочника. Причиной этого является то, что значения в справочнике для **VOH** определены при условии максимальной **IOH** загрузки. В цифровом SimCode коде **VOL** и **VOH** представляют напряжение ненагруженного выхода.

Для использования описываемой функции необходимо определить выводы питания в операторе **PWR_GND_PINS**. Значения **vol_param** и **voh_param**, установленные с помощью оператора **VOL_VOH_MIN**, перекрывают значения, определенные в диалоговом окне **Digital Model Parameters** в полях **VOL value** и **VOH value**. В операторе определяются значения смещений, а не абсолютные значения. Параметр **<voh offset>** должен быть отрицателен, чтобы при добавлении его к **pwr_param** полученное значение **VOH** не превышало **pwr_param**. Если разница между получившимися значениями **vol_param** и **voh_param** меньше **<min voh-vol>**, тогда **vol_param** принимает значение равно **gnd_param**, а **voh_param** будет равен **gnd_param + <min voh-vol>**.

Пример

```
VOL_VOH_MIN(0.2, -0.4, 0.1);
```

В этом примере:

- 1 Если `gnd_param = 0V` и `pwr_param = 5.0V`, то `vol_param = 0.2V` и `voh_param = 4.6V`
- 2 Если `gnd_param = 0V` и `pwr_param = 0.5V`, то `vol_param = 0.0V` и `voh_param = 0.1V`

Смотрите также

REALS, PWR_GND_PINS

WHILE ... DO

Циклическое выполнение в SimCode с выходом по условию.

Основная форма

```
WHILE (<expression>) DO BEGIN ... END;
```

Параметры

<expression> Выражение, которое может быть вычислено как истинное или ложное.

Применение

Оператор **WHILE ... DO** используется для выполнения в цикле части SimCode кода до тех пор, пока <expression> не примет ложного значения.

Замечания

Выполнение программы проходит в пределах цикла, обозначенного операторами **BEGIN** и **END**, до тех пор, пока <expression> не примет ложного значения. Затем выполнение программы продолжится со строки, следующей сразу за оператором **END**.

Examples

```
i = 1;
WHILE (i <= 5) DO
  BEGIN
    data[i] = data[i + 1];
    i = i + 1;
  END;
```

Смотрите также

IF ... THEN

WIDTH

Проверяет минимальную ширину импульсов на входных выводах.

Основная форма

```
WIDTH(<input> [<input>...] {TWL=<time>}{TWH=<time>} [<message>];
```

Параметры

<code><input></code>	Имя или индексная переменная входного вывода.
<code>TWL</code>	Ширина импульса в низком состоянии.
<code>TWH</code>	Ширина импульса в высоком состоянии.
<code><message></code>	Текстовая строка, которая выводится в качестве предупреждения.

Применение

Функция **WIDTH** сравнивает ширину импульса на каждом выводе `<input>` со временем, указанным в операторе. Время для сигналов в низком уровне определяется с помощью параметра **TWL=<time>**, а для высокого — **TWH=<time>**. Если сравниваемое время меньше указанного в `<time>`, то выводится соответствующее предупреждение. Если в операторе присутствует необязательный параметр `<message>`, то его содержимое будет использовано в качестве предупреждения.

Замечания

С описываемой функцией используются справочные данные. Входные выводы могут быть определены либо с помощью своих имен, либо посредством переменной целого типа, которая содержит индекс входного вывода. Имена и переменные одновременно могут использоваться в одной команде **WIDTH**.

Examples

```
WIDTH(CLK TWL=clk_twl TWH=clk_twh "CLK");  
WIDTH(PRE CLR TWL= pre_clr_twl "PRE or CLR");
```

WIDTH_TIME

Возвращает ширину последнего импульса, встретившегося на указанном выводе.

Основная форма

```
WIDTH_TIME(<input>)
```

Параметры

`<input>` Имя или индексная переменная входного вывода.

Применение

Эта функция возвращает значение вещественного типа, показывающее ширину последнего встретившегося на указанном в параметре `<input>` выводе импульса.

Пример

```
PW = (WIDTH_TIME(CP2));
```


Глава 5

Проектирование устройств на базе ПЛИС

Основные особенности проектирования устройств на базе ПЛИС

Данная глава описывает процесс проектирования устройств на базе программируемых логических устройств (ПЛИС, PLD). После изучения этой главы разработчик получит всю необходимую информацию по проектированию устройств на ПЛИС либо с использованием редактора принципиальных схем, либо с использованием языка описаний аппаратуры CUPL (рис. 5.1).

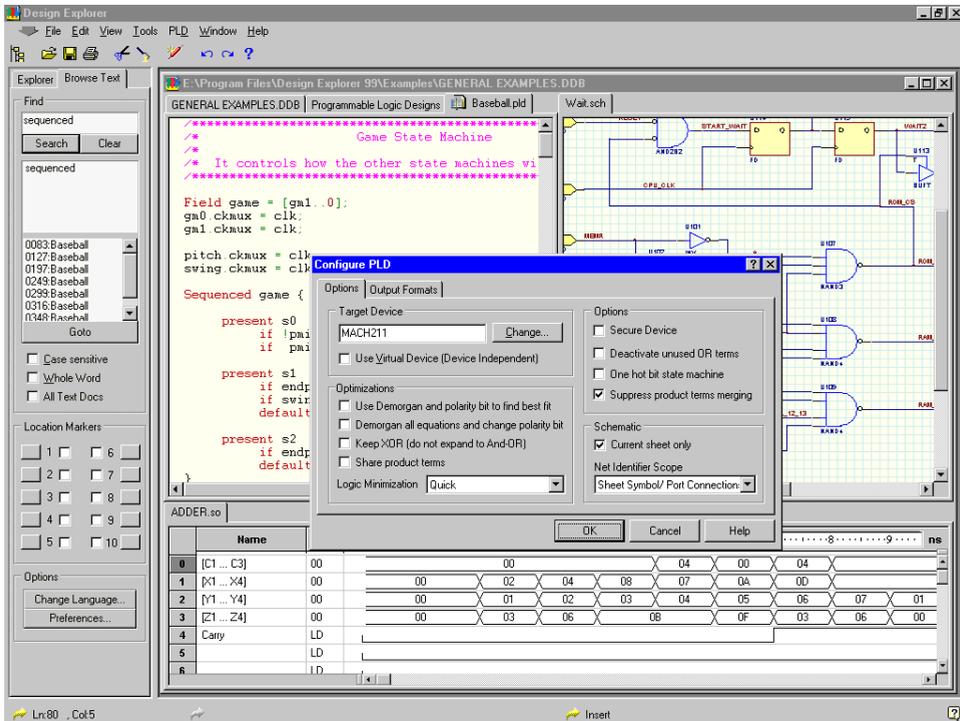


Рис. 5.1. Вид среды Design Explorer при проектировании устройств на базе ПЛИС.

Язык CUPL HDL позволяет программировать ПЛИС при помощи булевой алгебры, метода конечных автоматов и таблиц истинности. Разделы *Пример сессии разработки проекта на базе ПЛИС* и *Примеры проектов на основе ПЛИС* дадут пользователю исчерпывающее представление о процессе проектирования таких устройств.

В системе Protel 99 SE имеются обширные библиотеки элементов, поддерживающие устройства всех известных производителей программируемой логики. Язык CUPL HDL является независимым от производителя, дает разработчику значительную свободу в выборе элементной базы.

Описание проектов на базе ПЛИС с помощью принципиальных схем

Разработчик может зарисовать содержимое кристалла микросхемы ПЛИС как принципиальную схему с использованием специализированной библиотеки элементов **PLD Symbols Library**, в которой содержатся все необходимые для этого функциональные блоки. Проект такого устройства может быть как одно-, так многолистовым с иерархией различной степени сложности и глубины. Библиотека элементов **PLD Symbols.lib** находится в библиотечной базе данных с названием **Pld.Ddb**.

Описание проектов на базе ПЛИС с помощью языка высокого уровня

Файлы логического описания на языке CUPL HDL, а также файлы перечисления инструкций моделирования создаются при помощи встроенного текстового редактора, допускающего проверку синтаксиса. Все инструменты проектирования устройств на ПЛИС имеют встроенную функцию поиска ошибок и предупреждения о них. При обнаружении ошибок во время компиляции или моделировании проекта строка описания с ошибочным условием автоматически выводится на экран и подсвечивается.

Трассировщик-компилятор ПЛИС

Трассировщик-компилятор ПЛИС содержит самый быстрый и самый мощный минимизатор, предназначенный для упаковки логических схем, и имеющий четыре уровня минимизации. Компилятор упрощает логические выражения с использованием распределенных характеристик и теоремы ДеМоргана.

На выходе компилятора создается файл промышленного стандарта JEDEC, который совместим с любым программатором, поддерживающим этот формат. Формат поддерживает микросхемы большинства всех основных производителей программируемой логики, обеспечивая разработчику свободу в выборе базового кристалла.

Программа моделирования устройств на базе ПЛИС

До окончательного программирования кристалла пользователю необходимо провести его моделирование, что позволяет сделать программа **PLD Simulator**. Первоначально создается файл инструкций моделирования, описывающий предполагаемое функ-

ционирование ПЛИС в терминах входных и выходных значений. Программа моделирования сравнивает предполагаемые значения с рассчитанными во время компиляции. Тестовый вектор, проверенный программой моделирования можно включить в JEDEC файл и загрузить в логический программатор.

Просмотр временных диаграмм, полученных в результате моделирования

Результаты моделирования можно просматривать в редакторе временных диаграмм (Waveform Editor). Редактор считывает выходной файл с расширением **.SO**, генерируемый программой моделирования и отображает временную диаграмму в окне в стиле электронной таблицы.

Язык высокого уровня CUPЛ

Трасировщик-компилятор ПЛИС поддерживает язык высокого уровня CUPЛ Hardware Description Language. В этой книге он называется CUPЛ или язык CUPЛ.

Для сокращения времени моделирования язык CUPЛ содержит замену выражений на уравнения и стенографическое обозначение списков, адресных диапазонов и битовых полей.

Синтаксис конечных автоматов предоставляет средства реализации любого синхронного приложения, с использованием модели конечного автомата либо Миля (Mealy), либо Мура (Moore). Синтаксис таблиц истинности открывает путь для четкого выражения определенных логических конструкций.

Справочная система пакета Protel 99 SE содержит раздел CUPЛ Language Reference, содержащий исчерпывающую информацию по использованию этого языка.

Поддерживаемые устройства

Поддержка программируемых логических устройств различных производителей дает пользователям два ярко выраженных преимущества. Первое заключается в том, что пользователю необходимо изучить только среду проектирования и язык. Можно сконструировать все, что угодно: от простого адресного декодера на основе GAL16V8, вплоть до запатентованной конструкции на основе кристаллов серии 5000 фирмы Xilinx. Второе преимущество состоит в возможности упаковки одной и той же функциональной логики в физически различные микросхемы, что, несомненно, дает свободу в выборе производителя кристалла.

Обзор процесса проектирования устройств на базе ПЛИС

В этом разделе приведено описание процесса конструирования устройств на базе ПЛИС. Здесь очерчивается последовательность прохождения пользователя от построения концепции проекта до получения запрограммированного устройства.

Ввод проекта

Проект можно ввести либо при помощи редактора принципиальных схем, либо описать на языке CUPL HDL. Выбор метода будет зависеть от природы проекта и предпочтений разработчика.

Проект на основе принципиальной схемы

Для создания принципиальной схемы проекта на ПЛИС пользователь должен пользоваться компонентами из специализированной библиотеки **PLD Symbols.lib** базы данных **Pld.ddb**, находящейся в каталоге `\Program Files\Design Explorer 99 SE\Library\Sch`.

После завершения логического конструирования пользователю необходимо выбрать целевое устройство (конкретный тип микросхемы), сконфигурировать трассировщик-компилятор и включить опцию вывода в диалоговом окне `Configure`. Проекты на основе принципиальной схемы сначала автоматически транслируются компилятором в код CUPL HDL, а затем CUPL код компилируется обычным способом.

Проект, описанный на языке CUPL HDL

Аппаратно-ориентированный описательный язык CUPL позволяет разрабатывать проекты на базе ПЛИС различными методами, включая логические выражения, метод конечных автоматов и таблицы истинности. Для сокращения времени моделирования язык CUPL содержит замену выражений на уравнения и стенографическое обозначение списков, адресных диапазонов и битовых полей.

Синтаксис конечных автоматов предоставляет средства реализации любого синхронного приложения, с использованием модели конечного автомата либо Милля (Mealy), либо Мура (Moore). Синтаксис таблиц истинности открывает путь для четкого выражения определенных логических конструкций.

Компиляция проекта

При компиляции CUPL кода выполняются следующие шаги.

Предварительная обработка входного файла

При считывании входного файла программа компиляции обрабатывает препроцессорные директивы, такие как **\$DEFINE** и т. д. Затем генерируется промежуточный файл с дополнительно расшифрованными препроцессорными директивами.

Синтаксический анализ и преобразование конечных автоматов

Промежуточный файл анализируется табличным синтаксическим анализатором, формирующим таблицу перекодировки символов и подробные уравнения. Анализатор разлагает синтаксическую структуру конечных автоматов, таблицы истинности и функции пользователя на логические уравнения, а также во время обработки строк операторов выполняет элементарное логическое упрощение.

Проверка совместимости устройств и "деМорганизация"

Схема расположения выводов файла источника сличается с эталонной цоколевкой устройства из библиотечного файла с расширением **DL**, при этом проверяется, все ли используемые переменные файла источника корректно вписываются в архитектуру выбранного кристалла.

Кроме того, этот модуль программы компиляции применяет теорему ДеМоргана к выходным переменным, если их полярность не соответствует полярности тех или иных выводов микросхемы. Это дает возможность проектам с активными сигналами высокого уровня быть реализованным на микросхемах, где активными являются сигналы низкого уровня, и наоборот. К тому же на этой стадии компиляции выполняется первый уровень минимизации. Он уничтожит избыточные нулевые и единичные термы и термы произведения, содержащиеся внутри других термов произведения. На этом шаге строится конечная таблица символов, содержащая связи модели устройства и битовое представление логической структуры.

Логическая минимизация

Модуль логической минимизации выполняет алгоритмы минимизации битового потока логической структуры, полученного на предыдущем шаге. Он обрабатывает только те уравнения, которые требуют упрощения.

Модуль минимизации обрабатывает исходный файл выражений с помощью одного из выбранных алгоритмов: **Quick** (быстрый), **QuineMcCluskey**, **Presto** и **Espresso**. Лучшим алгоритмом для PAL архитектуры является **QuineMcCluskey**, а для PLA архитектуры — **Espresso** (очень хорош для совместно использующихся термов произведения). В общем случае, **Espresso** представляет собой достаточно быстрый алгоритм и дает почти те же результаты, что и **QuineMcCluskey**.

Сборка и формирование выходного файла

Сборочный модуль осуществляет упаковку проекта в требуемое устройство. Для этого он использует встроенные алгоритмы или вызывает внешнюю программу. Для того, чтобы передать булеву логику в микросхему, сборочный модуль управляет макроячейками и мультиплексорами сложных устройств. Если проект невозможно

упаковать в микросхему, появляется сообщение об ошибке. Кроме того, сборочный модуль формирует выходной файл, определенный в диалоговом окне **Configure**.

Именно сборочный модуль определяет, можно ли проект передать в кристалл микросхемы, и создает карту пережигания. Карта пережигания и символьная таблица используется в дальнейшем для выпуска документации и формирования JEDEC файлов.

Моделирование проекта

Для того, чтобы промоделировать проект сначала необходимо создать входной файл, он имеет расширение **SI**, а сам проект скомпилировать с включенной опцией **ABS output**.

Входной файл описывает предполагаемое функционирование устройства на ПЛИС в терминах входных и выходных значений (то есть зависимость выходных сигналов от входных). Программа моделирования сравнивает предполагаемые значения с рассчитанными во время компиляции, содержащимися в **ABS** файле. Тестовый вектор, проверенный программой моделирования можно включить в JEDEC файл и загрузить в логический программатор.

Программа моделирует задержки распространения, которые наиболее полно отражают поведение проекта после реализации в реальном кристалле микросхемы. Некоторые микросхемы имеют определенные особенности, и внутренние задержки для них смоделировать невозможно. Кроме этого, программа моделирования покажет реальное состояние гонок сигналов для этой микросхемы.

Где настраивается и запускается компилятор ПЛИС

Все установки проекта, а именно выбор кристалла микросхемы, опции компилятора и опции выхода настраиваются в диалоговом окне **Configure PLD**. Это окно выводится на экран выбором команды **PLD » Configure** из меню текстового редактора или же редактора принципиальных схем. Кроме того, настроить проект, провести его компиляцию и выполнить его моделирование можно из панели инструментов **PLDTools** (рис 5.2). На экран панель вызывается из меню выбором строки **View » PLD Toolbar**.

Компилятор запускается из меню выбором команды **PLD » Compile** или нажатием кнопки **Compile**. Программа моделирования запускается из меню выбором команды **PLD » Simulate** или нажатием кнопки **Simulate**.

Более подробно все это описано в разделе *Компиляция проекта на ПЛИС*.

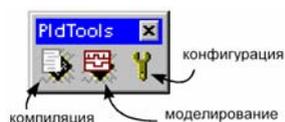


Рис 5.2. Панель управления PLDTools

Проектирование устройств на ПЛИС на основе принципиальных схем

Принципиальные схемы устройств на программируемой логике создаются также как и все остальные схемы на основе дискретных элементов размещаемых на печатной плате. Проект может состоять из одного или нескольких листов с неограниченной глубиной и сложностью иерархии. Единственное отличие состоит в том, что символы для создания принципиальной схемы необходимо брать из специальной библиотеки PLD Symbols.

При компиляции проект на ПЛИС сначала транслируется из принципиальной схемы в исходный файл компилятора языка CUPL. Полученный CUPL файл затем компилируется с целью получения выходного файла.

Использование библиотеки символов

Специализированная библиотека символов, предназначенная для создания проекта на ПЛИС, содержится в библиотеке **PLD Symbols.lib**, находящейся в базе данных по адресу **Program Files\Design Explorer 99 SE\Library\Sch\Pld.ddb**.

Библиотека содержит все стандартные функциональные группы: буферы, компараторы, триггеры и т. д.

Для того чтобы правильно выбрать из библиотеки элементы, реализуемые в той или иной микросхеме ПЛИС, необходимо хорошее знание разработчиком ее архитектуры.

Условные обозначения символов

Элементы схем обозначаются двумя способами. Большинство из них обозначаются набором букв и цифр в соответствии с их функциональным назначением, например NOR2B1. Другие обозначаются именами ТТЛ устройств из стандартной серии 74. Им предшествует буква X, например X74_138 (рис 5.3).

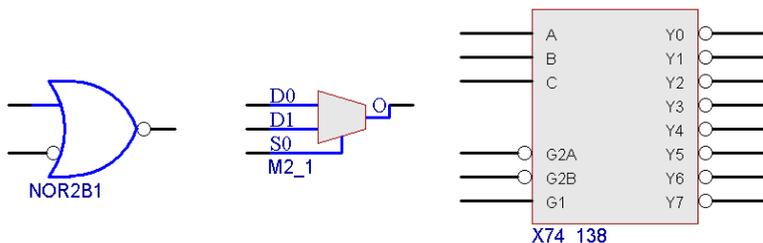


Рис 5.3. Вид некоторых элементов схем из библиотек ПЛИС

Поиск символов в библиотеке

Библиотека элементов программируемой логики содержит более 360 символов. Их список и подробное описание приведены в пункте *Элементы, сгруппированные по функциональным категориям*.

Для поиска нужного элемента можно использовать стандартные функции поиска редактора принципиальных схем. Для этого с помощью команды **Tools » Find Component** необходимо вызвать диалоговое окно **Find Component**. У каждого элемента библиотеки программируемой логики имеется подробное описание в поле Description, которое значительно облегчает его поиск. К примеру, разработчик желает знать, есть ли в библиотеке символ приоритетного шифратора с входами активного низкого уровня.

До начала поиска необходимо убедиться в правильности настроек области поиска

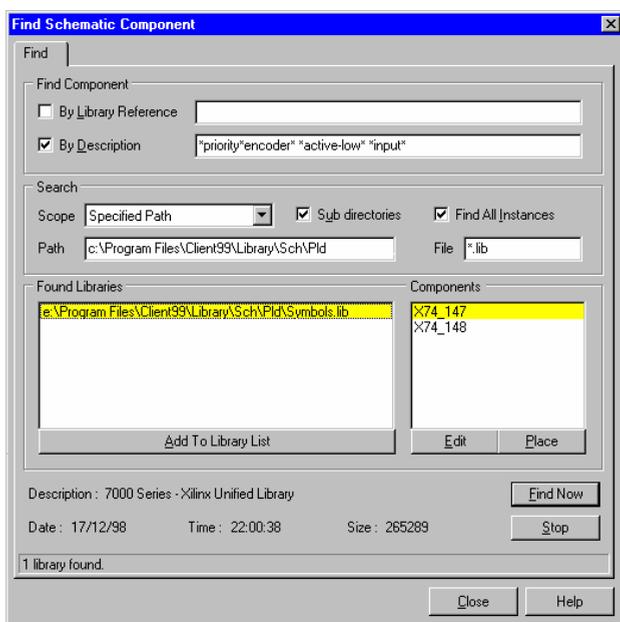


Рис. 5.4. Поиск нужного элемента производится стандартными средствами редактора схем

(**Search Scope**) и адреса библиотеки (**Search Path**). На рисунке 5.4 показано, как правильно задать каталог **\PLD**, чтобы ограничить область поиска библиотекой элементов программируемой логики.

Попробуем задать поисковую строку в точности же такую, какая может быть написана в библиотеке, заключая каждое отдельное слово или же набор слов в кавычки.

В разделе **Found Libraries** диалогового окна появится список тех библиотек, которые содержат эти слова в полях описания. Для просмотра поля описания достаточно выполнить щелчок левой кнопкой мыши на выбранном элементе.

Элементы, сгруппированные по функциональным категориям

Арифметические функции

В таблице 5.1 приведены три типа арифметических функций: аккумуляторы (ACC), сумматоры (ADD), и сумматоры/вычитатели (ADSU).

Таблица 5.1

Тип	Описание
ACC1	Одноразрядный каскадируемый аккумулятор с входом и выходом переноса и синхронным сбросом.
ACC16	Шестнадцатиразрядный каскадируемый аккумулятор с входом и выходом переноса и синхронным сбросом.
ACC4	Четырехразрядный каскадируемый аккумулятор с входом и выходом переноса и синхронным сбросом.
ACC8	Восьмиразрядный каскадируемый аккумулятор с входом и выходом переноса и синхронным сбросом.
ADD1	Одноразрядный полный сумматор с входом и выходом переноса.
ADD16	Шестнадцатиразрядный полный сумматор с входом и выходом переноса.
ADD4	Четырехразрядный полный сумматор с входом и выходом переноса.
ADD8	Восьмиразрядный полный сумматор с входом и выходом переноса.
ADSU1	Одноразрядный каскадируемый сумматор/вычитатель с входом и выходом переноса.
ADSU16	Шестнадцатиразрядный каскадируемый сумматор/вычитатель с входом и выходом переноса.
ADSU4	Четырехразрядный каскадируемый сумматор/вычитатель с входом и выходом переноса.
ADSU8	Восьмиразрядный каскадируемый сумматор/вычитатель с входом и выходом переноса.
X74_280	Девятиразрядный формирователь/контроллер четности/нечетности
X74_283	Четырехразрядный полный сумматор с входом и выходом переноса.

Буферные элементы

Буферы служат для ветвления схемы, для организации выхода с тремя состояниями, а также тактирования ПЛИС.

Таблица 5.2

Тип	Описание
BUF	Универсальный буфер
BUF16	16 универсальных буферов
BUF4	4 универсальных буфера
BUF8	8 универсальных буферов
BUFE	Внутренний буфер с тремя состояниями
BUFE16	16 внутренних буферов с тремя состояниями
BUFE4	4 внутренних буфера с тремя состояниями
BUFE8	8 внутренних буферов с тремя состояниями
BUFG	Глобальный тактовый буфер
BUFGP	Первичный глобальный буфер для управления тактовой шиной или длинными линиями (по 4 на ПЛИС)
BUFGS	Вторичный буфер для управления тактовой шиной или длинными линиями (по 4 на ПЛИС)
BUFT	Внутренний буфер с тремя состояниями
BUFT16	16 внутренних буферов с тремя состояниями
BUFT4	4 внутренних буфера с тремя состояниями
BUFT8	8 внутренних буферов с тремя состояниями

Компараторы

В таблице 5.3 приведены два типа компараторов: тождественности (**COMP**) и уровня (**COMPM**).

Таблица 5.3

Тип	Описание
COMP16	16-разрядный компаратор тождественности
COMP2	2-разрядный компаратор тождественности
COMP4	4-разрядный компаратор тождественности
COMP8	8-разрядный компаратор тождественности
COMPM16	16-разрядный компаратор уровня
COMPM2	2-разрядный компаратор уровня
COMPM4	4-разрядный компаратор уровня
COMPM8	8-разрядный компаратор уровня
COMPMC16	16-разрядный компаратор уровня
COMPMC8	8-разрядный компаратор уровня
X74_518	8-разрядный компаратор тождественности с разрешающим сигналом низкого уровня
X74_521	8-разрядный компаратор тождественности с разрешающим сигналом низкого уровня и активным низким уровнем выходного сигнала
X74_L85	4-разрядный расширяемый компаратор уровня

Счетчики

В таблице 5.4 представлены шесть типов счетчиков с различными модификациями синхронных и асинхронных входов. Информация о функциональном назначении счетчика заключена в его имени.

Таблица 5.4

Тип	Описание
CB16CE	16-разрядный каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB16CLE	16-разрядный загружаемый каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB16CLED	16-разрядный загружаемый каскадируемый реверсивный двоичный счетчик с разрешением счета и асинхронным сбросом.
CB16RE	16-разрядный каскадируемый двоичный счетчик с разрешением счета и синхронным сбросом
CB2CE	2-разрядный каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB2CLE	2-разрядный загружаемый каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB2CLED	2-разрядный загружаемый каскадируемый реверсивный двоичный счетчик с разрешением счета и асинхронным сбросом.
CB2RE	2-разрядный каскадируемый двоичный счетчик с разрешением счета и синхронным сбросом
CB4CE	4-разрядный каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB4CLE	4-разрядный загружаемый каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB4CLED	4-разрядный загружаемый каскадируемый реверсивный двоичный счетчик с разрешением счета и асинхронным сбросом.
CB4RE	4-разрядный каскадируемый двоичный счетчик с разрешением счета и синхронным сбросом
CB8CE	8-разрядный каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB8CLE	8-разрядный загружаемый каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CB8CLED	8-разрядный загружаемый каскадируемый реверсивный двоичный счетчик с разрешением счета и асинхронным сбросом.
CB8RE	8-разрядный каскадируемый двоичный счетчик с разрешением счета и синхронным сбросом
CC16CE	16-разрядный каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CC16CLE	16-разрядный загружаемый каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CC16CLED	16-разрядный загружаемый каскадируемый реверсивный двоичный счетчик с разрешением счета и асинхронным сбросом.

Таблица 5.4. Продолжение

Тип	Описание
CC16RE	16-разрядный каскадируемый двоичный счетчик с разрешением счета и синхронным сбросом
CC8CE	8-разрядный каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CC8CLE	8-разрядный загружаемый каскадируемый двоичный счетчик с разрешением счета и асинхронным сбросом.
CC8CLED	8-разрядный загружаемый каскадируемый реверсивный двоичный счетчик с разрешением счета и асинхронным сбросом.
CC8RE	8-разрядный каскадируемый двоичный счетчик с разрешением счета и синхронным сбросом
CD4CE	4-разрядный каскадируемый двоично-десятичный счетчик с разрешением счета и асинхронным сбросом.
CD4CLE	4-разрядный загружаемый каскадируемый двоично-десятичный счетчик с разрешением счета и асинхронным сбросом.
CD4RE	4-разрядный каскадируемый двоично-десятичный счетчик с разрешением счета и синхронным сбросом
CD4RLE	4-разрядный загружаемый каскадируемый двоично-десятичный счетчик с разрешением счета и синхронным сбросом.
CJ4CE	4-разрядный счетчик Джонсона с разрешением счета и асинхронным сбросом.
CJ4RE	4-разрядный счетчик Джонсона с разрешением счета и синхронным сбросом
CJ5CE	5-разрядный счетчик Джонсона с разрешением счета и асинхронным сбросом.
CJ5RE	5-разрядный счетчик Джонсона с разрешением счета и синхронным сбросом
CJ8CE	8-разрядный счетчик Джонсона с разрешением счета и асинхронным сбросом
CJ8RE	8-разрядный счетчик Джонсона с разрешением счета и синхронным сбросом
CR16CE	16-разрядный счетчик со сквозным переносом, с разрешением счета, с асинхронным сбросом и тактированием по отрицательному фронту
CR8CE	8-разрядный счетчик со сквозным переносом, с разрешением счета, с асинхронным сбросом и тактированием по отрицательному фронту
X74_160	4-разрядный двоично-десятичный счетчик с разрешением переноса и параллельной загрузки, синхронным сбросом, с загрузкой по активному сигналу низкого уровня
X74_161	4-разрядный двоичный счетчик с разрешением переноса и параллельной загрузки, синхронным сбросом, с загрузкой по активному сигналу низкого уровня
X74_162	4-разрядный двоичный счетчик с разрешением переноса и параллельной загрузки, синхронным сбросом, с загрузкой по активному сигналу низкого уровня

Таблица 5.4. Продолжение

Тип	Описание
X74_163	4-разрядный двоичный счетчик с разрешением переноса и параллельной загрузки, синхронным сбросом, с загрузкой по активному сигналу низкого уровня
X74_168	4-разрядный двоично-десятичный реверсивный счетчик с разрешением параллельной загрузки и переноса, с загрузкой по сигналу низкого уровня
X74_390	4-разрядный двоично-десятичный/ двоично-пятеричный счетчик с тактированием по отрицательному фронту и асинхронным сбросом

Регистры данных

В таблице 5.5 приведены три типа функций регистров данных, в точности повторяющих ТТЛ элементы.

Таблица 5.5

Тип	Описание
X74_174	6-разрядный регистр данных с асинхронным сбросом по сигналу низкого уровня
X74_273	8-разрядный регистр данных с асинхронным сбросом по сигналу низкого уровня
X74_377	8-разрядный регистр данных с разрешением загрузки по сигналу низкого уровня

Дешифраторы

Дешифраторы с входом разрешения можно использовать в качестве мультиплексоров (таблица 5.6).

Таблица 5.6

Тип	Описание
D2_4E	Дешифратор/демультиплексор 2 линии в 4 с разрешающим входом
D3_8E	Дешифратор/демультиплексор 3 линии в 8 с разрешающим входом
D4_16E	Дешифратор/демультиплексор 4 линии в 16 с разрешающим входом
X74_42	Дешифратор двоично-десятичного кода в десятичный 4 линии в 10 с активными выходными сигналами низкого уровня
X74_138	Дешифратор/демультиплексор 3 линии в 8 с выходными сигналами низкого уровня и 3-мя разрешающими входами.
X74_139	Дешифратор/демультиплексор 2 линии в 4 с выходными сигналами низкого уровня и разрешающим входом.
X74_154	Дешифратор/демультиплексор 4 линии в 16 с выходными сигналами низкого уровня и 2-мя разрешающими входами.

Шифраторы

В таблице 5.5 приведены два типа шифраторов.

Таблица 5.7

Тип	Описание
X74_147	Приоритетный двоичнодесятичный шифратор 10 линий в 4 с входными и выходными сигналами низкого уровня.
X74_148	Приоритетный каскадируемый шифратор 8 линий в 3 с входными и выходными сигналами низкого уровня

Триггеры

Существуют три типа триггеров (D, J-K, R-S) с различными вариантами синхронных и асинхронных входов (таблица 5.8). Некоторые из них имеют инверсные тактовые входы и/или, в отличие от сигнала reset, могут быть установлены в единицу или ноль по глобальному сигналу установки/сброса.

Таблица 5.8

Тип	Описание
FD	D-триггер
FD_1	D-триггер с синхронизацией по отрицательному фронту
FD16CE	16-разрядный регистр данных с разрешением тактирования и асинхронным сбросом.
FD16RE	16-разрядный регистр данных с разрешением тактирования и синхронным сбросом.
FD4CE	4-разрядный регистр данных с разрешением тактирования и асинхронным сбросом.
FD4RE	4-разрядный регистр данных с разрешением тактирования и синхронным сбросом.
FD8CE	8-разрядный регистр данных с разрешением тактирования и асинхронным сбросом.
FD8RE	8-разрядный регистр данных с разрешением тактирования и синхронным сбросом.
FDC	D-триггер с асинхронным сбросом
FDC_1	D-триггер с асинхронным сбросом и синхронизацией по отрицательному фронту
FDCE	D-триггер с разрешением тактирования и асинхронным сбросом
FDCE_1	D-триггер с разрешением тактирования, асинхронным сбросом и синхронизацией по отрицательному фронту
FDCP	D-триггер с асинхронными сбросом и установкой в единицу
FDCPE	D-триггер с разрешением тактирования и асинхронными сбросом и установкой в единицу

Таблица 5.8. Продолжение

Тип	Описание
FDCS	D-триггер с асинхронными сбросом и установкой в единицу
FDP	D-триггер с асинхронной установкой в единицу
FDP_1	D-триггер с асинхронной установкой в единицу и синхронизацией по отрицательному фронту
FDPE	D-триггер с разрешением тактирования и асинхронной установкой в единицу
FDPE_1	D-триггер с разрешением тактирования, асинхронной установкой в единицу и синхронизацией по отрицательному фронту
FDR	D-триггер с синхронным сбросом
FDRE	D-триггер с синхронным сбросом и разрешением тактирования
FDRS	D-триггер с синхронным сбросом и синхронной установкой в единицу
FDRSE	D-триггер с синхронным сбросом и синхронной установкой в единицу, с разрешением тактирования
FDS	D-триггер с синхронной установкой в единицу
FDSE	D-триггер с синхронной установкой в единицу и разрешением тактирования
FDSR	D-триггер с синхронной установкой в единицу и синхронным сбросом
FDSRE	D-триггер с синхронной установкой в единицу и синхронным сбросом и разрешением тактирования
FJKC	J-K триггер с асинхронным сбросом
FJKCE	J-K триггер с асинхронным сбросом и разрешением тактирования
FJKCP	J-K триггер с асинхронным сбросом и с асинхронной установкой в единицу
FJKCPE	J-K триггер с асинхронным сбросом, с асинхронной установкой в единицу и разрешением тактирования
FJKCS	J-K триггер с асинхронными сбросом и с асинхронной установкой в единицу
FJKP	J-K триггер с асинхронной установкой в единицу
FJKPE	J-K триггер с асинхронной установкой в единицу и разрешением тактирования
FJKRSE	J-K триггер с синхронным сбросом и синхронной установкой в единицу
FJKSRE	J-K триггер с синхронной установкой в единицу и синхронным сбросом
FTC	T-триггер (счетный, переключаемый) с разрешением переключения и асинхронным сбросом
FTCE	T-триггер с разрешением переключения, асинхронным сбросом и разрешением тактирования
FTCLE	T-триггер с возможностью загрузки, разрешением переключения, асинхронным сбросом и разрешением тактирования
FTCP	T-триггер с разрешением переключения, с асинхронным сбросом и с асинхронной установкой в единицу
FTCPE	T-триггер с разрешением переключения, с асинхронным сбросом, с асинхронной установкой в единицу и разрешением тактирования

Таблица 5.8. Продолжение

Тип	Описание
FTCPLE	T-триггер с возможностью загрузки, разрешением переключения, с асинхронным сбросом, с асинхронной установкой в единицу и разрешением тактирования
FTCS	T-триггер с разрешением переключения, с асинхронным сбросом и с асинхронной установкой в единицу
FTP	T-триггер с разрешением переключения и с асинхронной предварительной установкой в единицу
FTPE	T-триггер с разрешением переключения и с асинхронной предварительной установкой в единицу и разрешением тактирования
FTPLE	T-триггер с возможностью загрузки, разрешением переключения, с асинхронной предварительной установкой в единицу, и разрешением тактирования
FTRSE	T-триггер с разрешением переключения, с синхронным сбросом, синхронной установкой в единицу и разрешением тактирования
FTRSLE	T-триггер с возможностью загрузки, разрешением переключения, с синхронной установкой в единицу, с синхронным сбросом и разрешением тактирования
FTRSRE	T-триггер с разрешением переключения, с синхронным сбросом и синхронной установкой в единицу и разрешением тактирования.
FTRSLE	T-триггер с возможностью загрузки, с разрешением переключения, с синхронным сбросом и синхронной установкой в единицу и разрешением тактирования

Триггеры входа/выхода

Триггеры входа/выхода конфигурируются в ИОВ-блоки (блоки ввода/вывода). Среди них есть триггеры, на выходах которых стоят буферные элементы с разрешением включения третьего состояния, триггеры с глобальными установками в единицу или ноль и триггеры с инверсным входом синхронизации (таблица 5.9).

Таблица 5.9

Тип	Описание
IFD	Входной D-триггер
IFD_1	Входной D-триггер с инверсным входом синхронизации
IFD16	16 входных D-триггеров
IFD4	4 входных D-триггера
IFD8	8 входных D-триггеров
OFD	Выходной D-триггер
OFD_1	Выходной D-триггер с инверсным входом синхронизации
OFD16	16 выходных D-триггеров

Таблица 5.9. Продолжение

Тип	Описание
OFD4	4 выходных D–триггера
OFD8	8 выходных D–триггеров
OFDE	Выходной D–триггер с выходным буфером, включающимся по сигналу высокого уровня
OFDE_1	Выходной D–триггер с выходным буфером, включающимся по сигналу высокого уровня и инверсным входом синхронизации
OFDE16	16 выходных D–триггеров с выходным буфером, включающимся по сигналу высокого уровня
OFDE4	4 выходных D–триггера с выходным буфером, включающимся по сигналу высокого уровня
OFDE8	8 выходных D–триггеров с выходным буфером, включающимся по сигналу высокого уровня
OFDT	Выходной D–триггер с выходным буфером, включающимся по сигналу низкого уровня, переходящий в z-состояние по сигналу высокого уровня
OFDT_1	Выходной D–триггер с выходным буфером, включающимся по сигналу низкого уровня, переходящий в z-состояние по сигналу высокого уровня и инверсным входом синхронизации
OFDT16	16 выходных D–триггеров с выходным буфером, включающимся по сигналу низкого уровня, переходящим в z-состояние по сигналу высокого уровня
OFDT4	4 выходных D–триггера с выходным буфером, включающимся по сигналу низкого уровня, переходящим в z-состояние по сигналу высокого уровня
OFDT8	8 выходных D–триггеров с выходным буфером, включающимся по сигналу низкого уровня, переходящим в z-состояние по сигналу высокого уровня

Буферы ввода-вывода

В таблице 5.10 приведены входные и выходные буферные элементы с различными вариантами разрешающих сигналов по выходу.

Таблица 5.10

Тип	Описание
IBUF	Входной буфер
IBUF16	16 входных буферов
IBUF4	4 входных буфера
IBUF8	8 входных буферов
OBUF	Выходной буфер
OBUF16	16 выходных буферов
OBUF4	4 выходных буфера
OBUF8	8 выходных буферов

Таблица 5.10. Продолжение

Тип	Описание
OVUFE	Выходной буфер с тремя состояниями, включающийся по сигналу высокого уровня
OVUFE16	16 выходных буферов с тремя состояниями, включающихся по сигналу высокого уровня
OVUFE4	4 выходных буфера с тремя состояниями, включающихся по сигналу высокого уровня
OVUFE8	8 выходных буферов с тремя состояниями, включающихся по сигналу высокого уровня
OVUFT	Выходной буфер с тремя состояниями, включающийся по сигналу низкого уровня
OVUFT16	16 выходных буферов с тремя состояниями, включающихся по сигналу низкого уровня
OVUFT4	4 выходных буферов с тремя состояниями, включающихся по сигналу низкого уровня
OVUFT8	8 выходных буферов с тремя состояниями, включающихся по сигналу низкого уровня

Контакты ввода-вывода

Различные контакты, приведенные в таблице 5.11, служат для подключения к физическим выводам микросхем.

Таблица 5.11

Тип	Описание
IOPAD	Один контакт ввода/вывода сигналов
IOPAD16	16 контактов ввода/вывода сигналов
IOPAD4	4 контакта ввода/вывода сигналов
IOPAD8	8 контактов ввода/вывода сигналов
IPAD	Один контакт для ввода сигналов
IPAD16	16 контактов для ввода сигналов
IPAD4	4 контакта для ввода сигналов
IPAD8	8 контактов для ввода сигналов
OPAD	Один контакт для вывода сигналов
OPAD16	16 контактов для вывода сигналов
OPAD4	4 контакта для вывода сигналов
OPAD8	8 контактов для вывода сигналов

Входные триггеры-защелки

Входные триггеры-защелки, приведенные в таблице 5.12, используются для временного хранения данных проходящих на микросхему извне.

Таблица 5.12

Тип	Описание
ILD	Входной триггер–защелка сквозных данных
ILD_1	Входной триггер–защелка сквозных данных с инверсным стробирующим сигналом
ILD16	16 входных триггеров–защелок сквозных данных
ILD4	4 входных триггера–защелки сквозных данных
ILD8	8 входных триггеров–защелок сквозных данных

Инверторы

В таблице 5.13 приведены используемые логические инверторы.

Таблица 5.13

Тип	Описание
INV	Инвертор
INV16	16 инверторов
INV4	4 инвертора
INV8	8 инверторов

Триггеры-защелки

В таблице 5.14 приведены триггеры-защелки сквозных данных.

Таблица 5.14

Тип	Описание
LD	Триггер–защелка сквозных данных
LD_1	Триггер–защелка сквозных данных с инверсным стробирующим сигналом
LD16	16 триггеров–защелок сквозных данных
LD4	4 триггера–защелки сквозных данных
LD8	8 триггеров–защелок сквозных данных

Базовые логические элементы

Комбинаторные вентиляльные элементы, реализующие основные Булевы функции. В таблице 5.15 приведены элементы с количеством входов до 5 в комбинации с инверсными и не инверсными входами и до 9 входов с не инверсными входами.

Таблица 5.15

Тип	Описание
AND2	2-х входовой элемент И
AND2B1	2-х входовой элемент И с 1-м инверсным входом
AND2B2	2-х входовой элемент И с 2-мя инверсными входами
AND3	3-х входовой элемент И
AND3B1	3-х входовой элемент И с 1-м инверсным входом
AND3B2	3-х входовой элемент И с 2-мя инверсными входами
AND3B3	3-х входовой элемент И с 3-мя инверсными входами
AND4	4-х входовой элемент И
AND4B1	4-х входовой элемент И с 1-м инверсным входом
AND4B2	4-х входовой элемент И с 2-мя инверсными входами
AND4B3	4-х входовой элемент И с 3-мя инверсными входами
AND4B4	4-х входовой элемент И с 4-мя инверсными входами
AND5	5-ти входовой элемент И
AND5B1	5-ти входовой элемент И с 1-м инверсным входом
AND5B2	5-ти входовой элемент И с 2-мя инверсными входами
AND5B3	5-ти входовой элемент И с 3-мя инверсными входами
AND5B4	5-ти входовой элемент И с 4-мя инверсными входами
AND5B5	5-ти входовой элемент И с 5-ю инверсными входами
AND6	6-ти входовой элемент И
AND7	7-ми входовой элемент И
AND8	8-ми входовой элемент И
AND9	9-ти входовой элемент И
NAND2	2-х входовой элемент И-НЕ
NAND2B1	2-х входовой элемент И-НЕ с 1-м инверсным входом
NAND2B2	2-х входовой элемент И-НЕ с 2-мя инверсными входами
NAND3	3-х входовой элемент И-НЕ
NAND3B1	3-х входовой элемент И-НЕ с 1-м инверсным входом
NAND3B2	3-х входовой элемент И-НЕ с 2-мя инверсными входами
NAND3B3	3-х входовой элемент И-НЕ с 3-мя инверсными входами
NAND4	4-х входовой элемент И-НЕ
NAND4B1	4-х входовой элемент И-НЕ с 1-м инверсным входом
NAND4B2	4-х входовой элемент И-НЕ с 2-мя инверсными входами

Таблица 5.15. Продолжение

Тип	Описание
NAND4B3	4-х входовой элемент И-НЕ с 3-мя инверсными входами
NAND4B4	4-х входовой элемент И-НЕ с 4-мя инверсными входами
NAND5	5-ти входовой элемент И-НЕ
NAND5B1	5-ти входовой элемент И-НЕ с 1-м инверсным входом
NAND5B2	5-ти входовой элемент И-НЕ с 2-мя инверсными входами
NAND5B3	5-ти входовой элемент И-НЕ с 3-мя инверсными входами
NAND5B4	5-ти входовой элемент И-НЕ с 4-мя инверсными входами
NAND5B5	5-ти входовой элемент И-НЕ с 5-ю инверсными входами
NAND6	6-ти входовой элемент И-НЕ
NAND7	7-ми входовой элемент И-НЕ
NAND8	8-ми входовой элемент И-НЕ
NAND9	9-ти входовой элемент И-НЕ
NOR2	2-х входовой элемент ИЛИ-НЕ
NOR2B1	2-х входовой элемент ИЛИ-НЕ с 1-м инверсным входом
NOR2B2	2-х входовой элемент ИЛИ-НЕ с 2-мя инверсными входами
NOR3	3-х входовой элемент ИЛИ-НЕ
NOR3B1	3-х входовой элемент ИЛИ-НЕ с 1-м инверсным входом
NOR3B2	3-х входовой элемент ИЛИ-НЕ с 2-мя инверсными входами
NOR3B3	3-х входовой элемент ИЛИ-НЕ с 3-мя инверсными входами
NOR4	4-х входовой элемент ИЛИ-НЕ
NOR4B1	4-х входовой элемент ИЛИ-НЕ с 1-м инверсным входом
NOR4B2	4-х входовой элемент ИЛИ-НЕ с 2-мя инверсными входами
NOR4B3	4-х входовой элемент ИЛИ-НЕ с 3-мя инверсными входами
NOR4B4	4-х входовой элемент ИЛИ-НЕ с 4-мя инверсными входами
NOR5	5-ти входовой элемент ИЛИ-НЕ
NOR5B1	5-ти входовой элемент ИЛИ-НЕ с 1-м инверсным входом
NOR5B2	5-ти входовой элемент ИЛИ-НЕ с 2-мя инверсными входами
NOR5B3	5-ти входовой элемент ИЛИ-НЕ с 3-мя инверсными входами
NOR5B4	5-ти входовой элемент ИЛИ-НЕ с 4-мя инверсными входами
NOR5B5	5-ти входовой элемент ИЛИ-НЕ с 5-ю инверсными входами
NOR6	6-ти входовой элемент ИЛИ-НЕ
NOR7	7-ми входовой элемент ИЛИ-НЕ
NOR8	8-ми входовой элемент ИЛИ-НЕ
NOR9	9-ти входовой элемент ИЛИ-НЕ
OR2	2-х входовой элемент ИЛИ

Таблица 5.15. Продолжение

Тип	Описание
OR2B1	2-х входовой элемент ИЛИ с 1-м инверсным входом
OR2B2	2-х входовой элемент ИЛИ с 2-мя инверсными входами
OR3	3-х входовой элемент ИЛИ
OR3B1	3-х входовой элемент ИЛИ с 1-м инверсным входом
OR3B2	3-х входовой элемент ИЛИ с 2-мя инверсными входами
OR3B3	3-х входовой элемент ИЛИ с 3-мя инверсными входами
OR4	4-х входовой элемент ИЛИ
OR4B1	4-х входовой элемент ИЛИ с 1-м инверсным входом
OR4B2	4-х входовой элемент ИЛИ с 2-мя инверсными входами
OR4B3	4-х входовой элемент ИЛИ с 3-мя инверсными входами
OR4B4	4-х входовой элемент ИЛИ с 4-мя инверсными входами
OR5	5-ти входовой элемент ИЛИ
OR5B1	5-ти входовой элемент ИЛИ с 1-м инверсным входом
OR5B2	5-ти входовой элемент ИЛИ с 2-мя инверсными входами
OR5B3	5-ти входовой элемент ИЛИ с 3-мя инверсными входами
OR5B4	5-ти входовой элемент ИЛИ с 4-мя инверсными входами
OR5B5	5-ти входовой элемент ИЛИ с 5-ю инверсными входами
OR6	6-ти входовой элемент ИЛИ
OR7	7-ми входовой элемент ИЛИ
OR8	8-ми входовой элемент ИЛИ
OR9	9-ти входовой элемент ИЛИ
SOP3	3-х входовой элемент суммирования
SOP3B1A	3-х входовой элемент суммирования с 1-м инверсным входом И
SOP3B1B	3-х входовой элемент суммирования с 1-м инверсным входом ИЛИ
SOP3B2A	3-х входовой элемент суммирования с 2-мя инверсными входами И
SOP3B2B	3-х входовой элемент суммирования с 1-м инверсным входом ИЛИ и 1-м инверсным входом И
SOP3B3	3-х входовой элемент суммирования со всеми инверсными входами
SOP4	4-х входовой элемент суммирования
SOP4B1	4-х входовой элемент суммирования с 1-м инверсным входом И
SOP4B2A	4-х входовой элемент суммирования с 2-мя инверсными входами И
SOP4B2B	4-х входовой элемент суммирования с 1-м инверсным входом И у каждого элемента И
SOP4B3	4-х входовой элемент суммирования с 3-мя инверсными входами
SOP4B4	4-х входовой элемент суммирования со всеми инверсными входами

Таблица 5.15. Продолжение

Тип	Описание
XNOR2	2-х входовой элемент исключающее ИЛИ-НЕ
XNOR3	3-х входовой элемент исключающее ИЛИ-НЕ
XNOR4	4-х входовой элемент исключающее ИЛИ-НЕ
XNOR5	5-ти входовой элемент исключающее ИЛИ-НЕ
XNOR6	6-ти входовой элемент исключающее ИЛИ-НЕ
XNOR7	7-ми входовой элемент исключающее ИЛИ-НЕ
XNOR8	8-ми входовой элемент исключающее ИЛИ-НЕ
XNOR9	9-ти входовой элемент исключающее ИЛИ-НЕ
XOR2	2-х входовой элемент исключающее ИЛИ
XOR3	3-х входовой элемент исключающее ИЛИ
XOR4	4-х входовой элемент исключающее ИЛИ
XOR5	5-ти входовой элемент исключающее ИЛИ
XOR6	6-ти входовой элемент исключающее ИЛИ
XOR7	7-ми входовой элемент исключающее ИЛИ
XOR8	8-ми входовой элемент исключающее ИЛИ
XOR9	9-ти входовой элемент исключающее ИЛИ

Мультиплексоры

В таблице 5.16 приведены мультиплексоры с количеством входов до 16, с различными вариантами разрешающих и выходных сигналов.

Таблица 5.16

Тип	Описание
M16_1E	Мультиплексоры 16 в 1 с входом разрешения
M2_1	Мультиплексоры 2 в 1
M2_1B1	Мультиплексоры 2 в 1 с инверсией D0
M2_1B2	Мультиплексоры 2 в 1 с инверсией D0 и D1
M2_1E	Мультиплексоры 2 в 1 с входом разрешения
M4_1E	Мультиплексоры 4 в 1 с входом разрешения
M8_1E	Мультиплексоры 8 в 1 с входом разрешения
X74_150	Мультиплексоры 16 линий в 1 с активными разрешающим и выходными сигналами низкого уровня
X74_151	Мультиплексоры 16 линий в 1 с разрешающим сигналом низкого уровня и комплементарными выходами
X74_152	Мультиплексоры 8 линий в 1 с активными низкого уровня выходными сигналами

Таблица 5.16. Продолжение

Тип	Описание
X74_153	Два мультиплексора 4 линии в 1 с активным низкого уровня сигналом включения и общими сигналами выбора входа
X74_154	Дешифратор/демультиплексор 4 линии в 16 с активными низкого уровня выходными сигналами и двумя сигналами включения (разрешения)
X74_157	Четыре мультиплексора 2 в 1 с общим сигналом выбора входа и с активным низкого уровня сигналом включения.
X74_158	Четыре мультиплексора 2 в 1 с общим сигналом выбора входа и с активным низкого уровня сигналом включения и активными низкого уровня выходными сигналами.
X74_298	Четыре мультиплексора 2 в 1 с запоминанием и синхронизацией по отрицательному (заднему) фронту.
X74_352	Два мультиплексора 4 линии в 1 с активным низкого уровня сигналом включения и активными низкого уровня выходными сигналами.

Сдвиговые регистры

В таблице 5.17 приведены сдвиговые регистры разной разрядности и различными вариантами сигналов включения и сброса.

Таблица 5.17

Тип	Описание
SR4CE	4-х разрядный последовательно-параллельный сдвиговый регистр с разрешением синхронизации (счета) и асинхронным сбросом.
SR4CLE	4-х разрядный последовательно-параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и асинхронным сбросом.
SR4CLED	4-х разрядный последовательно-параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и асинхронным сбросом.
SR4RE	4-х разрядный последовательно-параллельный сдвиговый регистр с разрешением синхронизации (счета) и синхронным сбросом.
SR4RLE	4-х разрядный последовательно-параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и синхронным сбросом.
SR4RLED	4-х разрядный последовательно-параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и синхронным сбросом.
SR8CE	8-ми разрядный последовательно-параллельный сдвиговый регистр с разрешением синхронизации (счета) и асинхронным сбросом.

Таблица 5.17. Продолжение

Тип	Описание
SR8CLE	8–ми разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и асинхронным сбросом.
SR8CLED	8–ми разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и асинхронным сбросом.
SR8RE	8–ми разрядный последовательно–параллельный сдвиговый регистр с разрешением синхронизации (счета) и синхронным сбросом.
SR8RLE	8–ми разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и синхронным сбросом.
SR8RLED	8–ми разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и синхронным сбросом.
SR16CE	16–ти разрядный последовательно–параллельный сдвиговый регистр с разрешением синхронизации (счета) и асинхронным сбросом.
SR16CLE	16–ти разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и асинхронным сбросом.
SR16CLED	16–ти разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и асинхронным сбросом.
SR16RE	16–ти разрядный последовательно–параллельный сдвиговый регистр с разрешением синхронизации (счета) и синхронным сбросом.
SR16RLE	16–ти разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и синхронным сбросом.
SR16RLED	16–ти разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой, с разрешением синхронизации (счета) и синхронным сбросом.
X74_164	8–ми разрядный последовательно–параллельный сдвиговый регистр асинхронным сбросом по сигналу низкого уровня.
X74_165S	8–ми разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой и с разрешением синхронизации (счета).
X74_194	4–х разрядный реверсивный последовательно–параллельный сдвиговый регистр с параллельной загрузкой.
X74_195	4–х разрядный последовательно–параллельный сдвиговый регистр с параллельной загрузкой.

Циклические сдвиговые регистры

Таблица 5.18

Тип	Описание
BRLSHFT4	4-х разрядный циклический регистр сдвига.
BRLSHFT8	8-ми разрядный циклический регистр сдвига.

Символы источников питания и заземления

Таблица 5.19

Тип	Описание
GND	Элемент соединения с сигнальной землей
VCC	Элемент соединения с источником питания

ОЗУ(RAM)

Таблица 5.20

Тип	Описание
RAM16X1	Статическое ОЗУ с организацией 16x1 бит

Как определяется функциональное назначение символа

У каждого символа библиотеки символов программируемой логики (PLD) есть описание его функционального поведения. Описания хранятся в файле определения символов (**Symbol Definition File**) **PLDNet.SDF**. Синтаксис, используемый этим файлом, описан в его заголовке.

Разработка схемы внутренней логики

Внутренняя логическая схема программируемого устройства разрабатывается по стандартным методикам проектирования цифровых устройств. При создании схемы необходимо лишь извлекать элементы из библиотеки элементов программируемых устройств и затем обычным путем соединять их проводниками.

При компиляции проекта ПЛИС на основе принципиальной схемы последний транслируется компилятором языка CUPL из схемы в файл исходного кода. Затем полученный файл компилируется в назначенные выходные файлы.

Общие замечания по проектированию ПЛИС:

- Каждый элемент в проекте, включая входные и выходные контакты, должен иметь уникальное позиционное обозначение. Особых требований или ограничений на идентификаторы не предъявляется.

- Для облегчения отладки и нахождения неполадок рекомендуется присваивать имена всем внутренним цепям.
- Имена внутренних цепей должны быть уникальны и не могут быть инвертированным значением любого другого имени цепи.
- Допускается использовать шины, причем, они должны иметь метки цепей стандартного для шин формата, диапазон обозначения может быть как возрастающий так и убывающий.
- Используйте те символы, которые поддерживаются для конкретного типа кристалла микросхемы.

Создание многолистного проекта

Многолистные проекты создаются аналогично проектам из одного листа принципиальной схемы. Редактор схем поддерживает ряд методов межлистовых соединений, доступных и для проектирования ПЛИС. Подробнее эти методы описаны в разделе *Управление многолистowymi и иерархическими проектами* главы *Разработка принципиальных схем*.

Метод межлистовых соединений определяется областью действия идентификаторов цепей, используемой при создании списка цепей. Для этого перед компиляцией проекта компилятором языка CUPL необходимо в диалоговом окне конфигурирования ПЛИС (**Configure PLD**), запускаемым командой меню **PLD » Configure**, выбрать заданную область действия идентификаторов цепей (**Net Identifier Scope**).

Самый простой способ запомнить, что означает каждая из областей действия идентификаторов цепей, следующий. Соединения **Sheet Symbol/Port** (символы листа/порты) создают вертикальные связи между портами и соответствующими точками входа листа схемы, в то время как два других варианта создают горизонтальные межлистовые соединения между соответствующими портами и метками цепей. Если разработчик планирует пользоваться многоуровневой иерархией листов принципиальной схемы, ему необходимо выбрать вариант соединения **Sheet Symbol/Port**.

Стыковка внутренней логической схемы с выводами компонента

Для установления связи внутренней логической схемы с выводами выбранной микросхемы ПЛИС предназначены специальные символы контактов ввода (**IPAD**), вывода (**OPAD**), ввода/вывода (**IOPAD**) сигналов.

Разработчику необходимо поместить соответствующий контакт (**PAD**) у каждой точки соединенной с выводом компонента и присвоить ему уникальное позиционное обозначение. Номер вывода микросхемы, соединенного с данным контактом указывается в поле **Part Field1** символа контакта следующим образом:

`LOC=PIN[номер вывода компонента]`

Использование многовыводных групп контактов

В библиотеке элементов имеется ряд многовыводных групп контактов типа **IPAD**, **OPAD** и **IOPAD**.

Таблица 5.21

Цепь	Вывод микросхемы
DBus0	21
DBus1	22
DBus2	23
DBus3	24
DBus4	25
DBus5	39
DBus6	38
DBus7	32

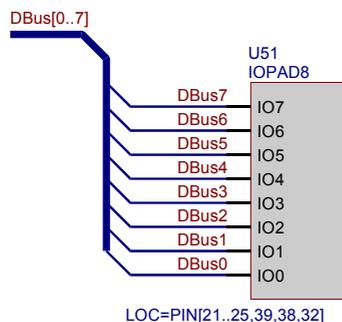


Рис. 5.5. Определение диапазонов контактов.

С их помощью можно подсоединять внутреннюю шину к выводам микросхемы. Для назначения соответствия контактов выводам реальной микросхемы, указываемой в поле **Part Field1**, можно использовать следующие синтаксические структуры: либо определять диапазон контактов, разделяя номера первого и последнего контакта двумя точками (по убыванию или возрастанию), либо определять их индивидуально, отделяя запятой друг от друга (рис. 5.5). Возможное распределение контактов для показанного на рисунке элемента **IOPAD8** отображено в таблице 5.21.

Отображение на схеме номеров выводов

Номера выводов можно отобразить на листе, выбрав в меню редактора схем команду **Simulate » Display Pin LOCs**. Чтобы вновь скрыть номера выводов, необходимо повторно выполнить эту команду меню.

Компиляция проекта ПЛИС на основе принципиальной схемы

После завершения разработки проекта требуется сконфигурировать компилятор языка CUPL, для чего необходимо выполнить команду меню **PLD » Configure**. Подробная информация по установке параметров оптимизации компилятора, его опциях и выходных форматах приведена в разделе *Компиляция проекта на ПЛИС*.

При компиляции многостраничных проектов необходимо в диалоговом окне **Configure PLD** отключить опцию **Current sheet only** (только текущий лист) и правильно задать область действия идентификаторов цепей (**Net Identifier Scope**).

При запуске процесса компиляции проект сначала из схемы транслируется в исходный файл компилятора языка CUPL. Затем полученный CUPL файл с расширением **PLD** компилируется для создания назначенных выходных файлов.

При компиляции проекта CUPL компилятором можно воспользоваться опцией виртуального устройства (**Virtual Device**). Она позволяет проверить, будет ли вообще компилироваться проект, а также определить требуемое необходимое максимальное количество комбинационных логических блоков (термов производства). Их число приводится в **DOC** файле. Большое число КЛБ в конечном счете может стать причиной

использования кристалла с неоправданно большим их объемом. Чем больше блоков необходимо для проекта, тем больше их должно быть в кристалле. Различные комбинации опций и параметров оптимизации компилятора позволяют сократить число требуемых комбинационных логических блоков.

Описание проекта устройства на ПЛИС на языке высокого уровня CUPL HDL

Исходный файл компилятора PLD представляет собой текстовый файл, содержащий логическое описание проекта на базе ПЛИС на языке Hardware Description Language CUPL. Этот файл является исходным для компилятора PLD, выполняющего компиляцию проекта и формирующего файл пригодный для загрузки в программатор.

При разработке проекта на ПЛИС необходимо:

Тщательно изучить задачу проектирования: При рассмотрении требований к проекту нужно помнить, что разработчику доступны различные методы проектирования или описания, а именно: метод конечных автоматов, логические уравнения и таблица истинности. Пользователю следует определить тот метод проектирования, который больше всего подходит для решения его задачи.

Создать исходный файл компилятора: При создании исходного файла необходимо пользоваться синтаксическими правилами, описанными в этой главе. Следует помнить, что при создании нового файла первым делом необходимо отредактировать его заголовок.

Сформулировать уравнения: Для точного определения разрабатываемой логической схемы уравнения должны быть записаны на языке CUPL, таблицей истинности, в Булевом формате или формате конечных автоматов.

Определить тип целевой микросхемы: Количество входных, а также зарегистрированных и незарегистрированных выходных выводов должно быть достаточным для реализации проекта. Выбранная микросхема должна иметь достаточное количество комбинационных логических блоков (КЛБ). При необходимости, рекомендуется проверить, имеет ли микросхема управление третьим состоянием выхода (z-состоянием). Выбор типа микросхемы осуществляется в диалоговом окне конфигурации проекта на базе ПЛИС (**Configure PLD**).

Распределить выводы микросхемы: Определите, какие выводы микросхемы будут соответствовать входам и выходам проекта. Информация по использованию каждой корректной микросхемы приводится в справочных материалах ее производителя.

Подготовить проект к компиляции: Разработчик должен определить форматы файлов для загрузки и моделирования и выбрать один из четырех доступных методов минимизации. После этого проект готов к компилированию. За информацией о настройках в диалоговом окне конфигурации проекта на базе ПЛИС (**Configure PLD**) следует обратиться к разделу *Компиляция проекта на ПЛИС*.

Обзор синтаксиса языка CUPL

При создании исходного файла для компилятора необходимо придерживаться общей линии, описанной ниже. В начале файла должен всегда располагаться заголовок, сразу за которым следует определения выводов и узлов. Остальные описываемые здесь информационные секции используются по мере необходимости и могут располагаться в произвольном порядке. В интерактивную справочную систему программы Protel 99 SE включена вся необходимая информация о языке CUPL.

Заголовок

Заголовок содержит справочную информацию о проекте. В нее входят: имя файла, тип устройства, дата начала разработки, номер версии проекта, название компании, имя разработчика, информация о блоке, где данное устройство используется, позиционное обозначение устройства в этом блоке. В заголовке также можно задать тип целевой микросхемы и тип выходного файла, формируемого компилятором. Все перечисленные поля заголовка, за исключением поля имени файла, являются необязательными и могут располагаться в произвольном порядке. Во время компиляции при отсутствии заголовка появится соответствующее предупреждение, но сам процесс прерываться не будет.

Name	XXXXXX;
Partno	XXXXXX;
Date	XX/XX/XX;
Revision	XX;
Designer	XXXXXX;
Company	XXXXXX;
Assembly	XXXXXX;
Location	XXXXXX;
Format	XXXXXX;
Device	XXXXXX;

Титульный блок

Титульный блок – это область строк комментариев, в которой разработчик может поместить название или описание проекта. Сюда же можно поместить название микросхемы и имя производителя для возможности дальнейшего использования.

```
/*
*****
/*
*****
/*
*****
/* Allowable Target Device Types:
*****
/*
```

Секция описания выводов и узлов

В секции определения выводов и узлов записываются операторы определения выводов, снабженные соответствующими комментариями. Описания выводов можно располагать в произвольном порядке, но для облегчения понимания проекта, их следует группировать по функциональному назначению. Пустая заготовка под комментарий после описания каждого вывода служит лишним напоминанием о необходимости ввода необходимой информации о предназначении вывода в проекте.

```

/** Inputs **/
Pin      =          ;          /*
Pin      =          ;          /*
Pin      =          ;          /*

/** Outputs **/
Pin      =          ;          /*
Pin      =          ;          /*
Pin      =          ;          /*

Pinnode  =          ;          /*
Pinnode  =          ;          /*
    
```

Вспомогательные переменные

Вспомогательные переменные используются для удобства формирования логических уравнений или других вспомогательных переменных. Написание логических уравнений способом "сверху вниз" делает файл описания логической схемы более легким для прочтения и понимания.

Логические выражения

Эта секция является главной частью исходного файла. Здесь записываются все необходимые части проекта, такие как конечные автоматы, таблицы истинности и булевы выражения.

Директивы исходного файла

Большинство директив исходного файла начинается со знака \$. Эти команды представляют собой команды для препроцессора. Директивы должны записываться с первой позиции любой строки исходного файла. Набирать команды на клавиатуре можно в любой комбинации заглавных или строчных букв.

Описание символьных констант

Символьные константы определяются командой \$define. Эта команда замещает строку символов другим специфическим оператором, числом или символом. Такая подстановка является замещением строки букв, сделанная входным файлом до того как

он будет обработан компилятором. Отменить определение символьной константы можно командой **\$undef**, которая команда нейтрализует действие директивы **\$define**.

Присоединение других файлов

Внешние файлы подключаются к текущему исходному файлу командой **\$include**. Присоединение файлов позволяет разработчику стандартизировать часть наиболее часто употребляемого кода. Эта функция также полезна для сохранения в отдельном файле параметров, где определяются константы используемые в различных исходных файлах. Присоединяемые файлы также могут содержать команды **\$include**, что допускает их многократное вложение.

Условная компиляция

Условная компиляция позволяет пользователю компилировать секции исходного кода в зависимости от наличия или отсутствия определяемых символьных констант. В число команд условной компиляции входят: **\$ifdef**, **\$ifndef**, **\$else**, **\$endif**. Если символьная константа предварительно определена, то операторы исходного кода, стоящие за командой **\$ifdef**, обрабатываются компилятором до момента появления команды **\$else** или **\$endif**. Если же константа не была определена, то все операторы, стоящие за командой **\$ifdef** игнорируются. При ложности проверяемых условий команд **\$ifdef** и **\$ifndef**, компилируются операторы, стоящие между директивами **\$else** или **\$endif**, в противном случае они игнорируются. Команда **\$endif** заканчивает область условной компиляции, начатой командами **\$ifdef** и **\$ifndef**. Условная компиляция может быть вложенной, каждый уровень вложенности должен обрамляться командами **\$ifdef** (**\$ifndef**) и **\$endif**.

```

$DEFINE Prototype X /* define Prototype*/
$IFDEF Prototype
pin 1           = memreq; /* memory request on */
                    /* pin 1 of prototype*/
pin 2           = ioreq; /* I/O request on*/
                    /* pin 2 of prototype*/
$ELSE
pin 1           = ioreq; /* I/O request on*/
                    /* pin 1 of PCB*/
pin 2           = memreq; /* memory request on */
                    /* pin 2 of PCB*/
$ENDIF
    
```

Операторы циклов

Для циклического исполнения участков кода языка CUPL существуют команды **\$repeat** и **\$repnd**. Эта команда аналогична команде FOR в языке C и команде DO в языке FORTRAN. Индекс команды **\$repeat** определяет количество повторений участка кода, находящегося между директивами **\$repeat** и **\$repnd**. Использование этих команд дает возможность очень легко описать счетчики на основе конечных автоматов.

Исходный CUPL код:

Результат после обработки препроцессором:

<pre>FIELD sel = [in2..0] \$REPEAT i = [0..7] !out{i} = sel:'h'{i} & enable; \$REPEND</pre>	<pre>FIELD sel = [in2..0]; !out0 = sel:'h'0 & enable; !out1 = sel:'h'1 & enable; !out2 = sel:'h'2 & enable; !out3 = sel:'h'3 & enable; !out4 = sel:'h'4 & enable; !out5 = sel:'h'5 & enable; !out6 = sel:'h'6 & enable; !out7 = sel:'h'7 & enable;</pre>
---	--

Использование макросов

Макросы представляют собой пользовательские процедуры, вызываемые упоминанием их имени. Макрос сопровождается операторами **\$macro** и **\$mend**. Код, находящийся между ними в общем случае при компиляции будет игнорироваться. Макрос вызывается помещением имени макроса в теле исходного файла и передачей ему соответствующих параметров. Макросы можно использовать для создания библиотек наиболее часто применяемых или стандартных устройств, например, декодеров, счетчиков и т. д. Чтобы увидеть как препроцессор обрабатывает определения макросов имеется возможность создать файл с расширением (**.MX**).

```
$MACRO decoder bits MY_X MY_Y MY_enable;
    FIELD select = [MY_Y{bits-1}..0];
    $REPEAT i = [0..{2**{bits-1}}]
        !MY_X{i} = select:'h'{i} & MY_enable;
    $REPEND
$MEND
_/* Other CUPL statements */
decoder(3, out, in, enable); /*macro function call*/
```

Индивидуальная минимизация выводов

Выходы можно минимизировать последовательно при помощи оператора **MIN**. Для оператора **MIN** доступны 5 уровней минимизации, аналогичных задаваемым в диалоговом окне **Configure PLD**. Этот оператор разрешает установить индивидуальный уровень минимизации для различных выводов одного и того же проекта.

Ниже в таблице 5.22 приведено соответствие уровней минимизации опциям диалогового окна **Configure PLD**.

Таблица 5.22

Опция минимизации	Уровень минимизации
0	None
1	Quick

2	Quine-McCluskey
3	Presto
4	Expresso

Ниже приведен пример, иллюстрирующий индивидуальную минимизацию выводов:

MIN async_out	= 0;	/* no reduction	*/
MIN [outa, outb]	= 2;	/* Quine-McCluskey reduction	*/
MIN count.d	= 4;	/* Expresso reduction	*/

Технологически-зависимые плавкие перемычки

Некоторые микросхемы имеют плавкие перемычки, которые могут изменить поведение всего устройства. Как правило эти перемычки называются битами **MISER** и **TURBO**. В зависимости от требуемого поведения микросхемы пользователь должен либо их пережечь либо оставить. Поведение микросхемы устанавливается оператором **FUSE**. При задании номера перемычки и ее значения компилятор установит эту перемычку в соответствующее состояние. Пользоваться этим оператором нужно крайне осторожно, так как его некорректное использование может привести к непредсказуемым результатам.

FUSE(101,1);	/* Blow the Turbo bit */
FUSE(102,0);	/* Do not blow the Miser bit */

Элементы языка CUPL

В этом разделе приводится описание основных элементов языка CUPL.

Определение выводов и узлов

Так как описания выводов должны быть объявлены в начале исходного файла, их задание естественным образом становится отправной точкой в разработке проекта. Узлы и узловы выводы, описывающие внутренние регистры, также должны быть объявлены в начале исходного файла. Если пользователь уже знает, какую микросхему он будет использовать для реализации проекта, то ему необходимо расписать ее выводы. Однако при использовании для разводки проекта виртуального устройства необходимо лишь задать имена переменных, которые в последствии будут распределены по соответствующим выводам, а поле, которое обычно содержит номер контакта, оставить пустым.

Задание вспомогательных переменных

Вспомогательные переменные представляют собой переменные, относящиеся к логическим уравнениям, а не к контактам или узлам. Они применяются для определе-

ния уравнений, часто используемых при расчете других переменных, а также для облегчения понимания проекта.

Применение индексных переменных

Имена переменных, оканчивающиеся на десятичное число от 0 до 31, относятся к индексным переменным. Они могут быть полезны для представления групп адресных линий (шин адреса), линий данных (шин данных), или других объектов с последовательной нумерацией. Цифра 0 в индексе переменной при операциях с битовыми полями всегда обозначает младший разряд слова.

Различные форматы представления чисел

Все действия с числами компилятор производит с 32-х разрядной точностью. Следовательно, числа могут принимать значения от 0 до 232-1. Числа могут быть представлены в одном из 4-х форматов: бинарном (двоичном), восьмеричном, десятичном и шестнадцатеричном. Основным форматом для всех чисел является шестнадцатеричный, за исключением номеров выводов и индексных переменных, которые всегда десятичные. Символ X в составе двоичных, восьмеричных, и шестнадцатеричных чисел указывает на неопределенное значение соответствующих разрядов. Примеры различного представления чисел приведены в таблице 5.23.

Таблица 5.23

Число	Формат	Десятичное значение
'b'0	Бинарный	0
'B'1101	Бинарный	13
'O'663	Восьмеричный	435
'D'92	Десятичный	92
'h'BA	Шестнадцатеричный	186
'O'[300..477]	Восьмеричный (диапазон)	192..314
'H'7FXX	Шестнадцатеричный (диапазон)	32512..32767

Применение списков

Использование списков дает возможность быстрого определения групп переменных. Он обычно используется при распределении выводов и узлов, описании битовых полей и операций над множествами. Для разграничения элементов списка используются квадратные скобки.

Использование битовых полей

При задании битового поля группе битов присваивается общее имя переменной. После назначения битового поля ключевым словом **FIELD**, его имя можно использовать в логических выражениях. При этом операция, определяемая в выражении, применяется к каждому биту в группе. Когда встречается оператор **FIELD**, компилятор автоматически формирует отдельное 32-х разрядное поле, в которое заносятся все отдельные биты. Если используется индексная переменная, то номер бита, представляюще-

го элемент битового поля, соответствует ее индексному номеру. Это означает, что A0 всегда будет занимать нулевой бит битового поля. Как правило, при помощи битовых полей управляют шинами адреса и данных.

```
FIELD ADDRESS = [A7, A6, A5, A4, A3, A2, A1, A0];
                or
FIELD ADDRESS = [A7..0];

FIELD Mode = [Up, Down, Hold];
```

Синтаксис языка

В этом разделе приводится описание логических и арифметических операторов и функций, необходимых при создании проектов с помощью языка CUPL.

Логические операторы

В таблице приведены четыре основных логических оператора **NOT**, **AND**, **OR** и **XOR** и их приоритет по отношению друг к другу от самого высокого к самому низкому.

Таблица 5.24

Оператор	Пример использования	Выполняемая функция	Приоритет
!	!A	NOT	1
&	A & B	AND	2
#	A # B	OR	3
\$	A \$ B	XOR	4

Арифметические операторы и функции

В таблице 5.25 ниже приведены шесть основных арифметических операторов, используемых в командах **\$repeat** и **\$macro**, и их приоритет по отношению друг к другу от самого высокого к самому низкому.

Таблица 5.25

Оператор	Пример использования	Выполняемая функция	Приоритет
**	2**3	Возведение в степень	1
*	2*I	Умножение	2
/	4/2	Деление	2
%	9%8	Модуль	2
+	2+4	Сложение	3
—	4—I	Разность	3

Для команд **\$repeat** и **\$macro** доступна только одна арифметическая функция — функция логарифма. В таблице 5.26 приведены примеры логарифмических функций для разных оснований логарифма.

Таблица 5.26

Функция	Основание
LOG2	2
LOG8	8
LOG16	16
LOG	10

Функция логарифма возвращает целое значение. Например:

$$\begin{aligned} \text{LOG2}(32) &= 5 \iff 2^{**}5 = 32 \\ \text{LOG2}(33) &= \text{ceil}(5.0444) = 6 \iff 2^{**}6 = 64 \end{aligned}$$

Функция **Ceil(x)** возвращает наименьшее целое число, превышающее аргумент **x**.

Переменные с расширениями

К именам переменных можно добавлять расширения, обозначающие специфические функции, ассоциированные с основными узлами внутри микросхемы, например, описание триггерных схем и программируемые разрешающие сигналы с тремя состояниями. Компилятор проверяет использование расширения, чтобы определить, обоснованно ли его применение для заданного типа микросхемы и нет ли конфликтов с другими расширениями. Расширения используются компилятором для конфигурирования макроячеек внутри кристалла. Благодаря этому, пользователю не нужно знать, чем именно в макроячейке управляют те или иные перемиычки.

Рисунок 5.6 иллюстрирует применение расширений для имен переменных. Заметим, что этот рисунок не является представлением реальной схемы, а только показывает применение расширений для записи выражений различных функций цепи.

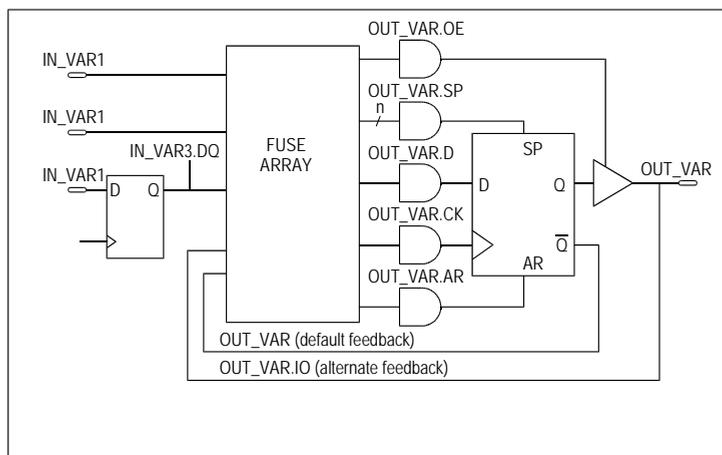


Рис. 5.6. Схема, иллюстрирующая использования расширений переменных

Логические выражения

Логические выражения представляют собой основные строительные блоки языка CUPL. Форма записи логических выражений следующая:

```
[!] var [.ext] = exp ;
```

где:

var — одна переменная или список индексированных или неиндексированных переменных, определенных в соответствии с правилами индексации списков. У списков переменных выражения определены для каждой переменной в списке.

.ext — необязательное расширение имени переменной, служащая для указания функции основных узлов программируемой микросхемы.

exp — выражение, то есть комбинация различных переменных и операторов.

Знак = является оператором присваивания. Он присваивает значение выражения переменной или множеству переменных.

Знак ! — оператор дополнения.

В стандартных логических уравнениях одной переменной обычно присваивается лишь одно выражение. Оператор **APPEND** позволяет одной переменной присвоить значения нескольких выражений. Значения дополнительных выражений взаимодействуют со значением первоначального выражения при помощи логической функции ИЛИ (**OR**). Формат использования оператора **APPEND** идентичен определению логического уравнения, за исключением того, что ключевое слово **APPEND** ставится непосредственно перед логическим выражением.

Логические выражения рекомендуется располагать в соответствующей секции исходного файла, как это показано в файле-шаблоне.

Применение операций над множествами

Все действия с однобитовой информацией, например контакт ввода, регистр, или выходной контакт можно превратить в многоразрядные, сгруппированные в наборы (множества). Множественные операции могут совершаться между множеством и переменной или выражением, а также между двумя множествами.

Результатом операции между множеством и одной переменной является новое множество, в котором действие выполняется между каждым элементом множества и указанной переменной.

Если действие выполняется между двумя множествами, то они должны иметь одинаковую размерность (то есть содержать одинаковое количество элементов). Результатом операций между двумя множествами является новое множество, в котором действия выполняются между элементами обоих множеств.

При использовании чисел в операциях над множествами, они (числа) представляются набором бинарных цифр. Восьмеричное число отображается набором из трех бинарных цифр, а десятичное или шестнадцатеричное — набором из четырех бинарных цифр.

Операции равенства

В отличие от операций над множествами операция равенства вычисляется для одного логического выражения. Она побитно проверяет равенство между множеством переменных и константой. Позиция бита константы сверяется с соответствующей позицией в множестве. При битовой позиции равной бинарной 1 (единице) элемент множества не изменяется. Если позиция бита равна 0 (нулю), элемент множества инвертируется. Когда же она равна бинарному X, элемент множества удаляется. Получившиеся элементы затем логически перемножаются складываются друг с другом, чтобы образовать единственное выражение.

Оператор тождественности также может использоваться с множеством одинаково обрабатываемых переменных. Например, три следующих выражения:

```
[A3, A2, A1, A0] : &
[B3, B2, B1, B0] : #
[C3, C2, C1, C0] : $
```

идентичны выражениям:

```
A3 & A2 & A1 & A0
B3 # B2 # B1 # B0
C3 $ C2 $ C1 $ C0
```

Действия с диапазонами значений

Операция с диапазонами значений похожа на операцию равенства, за исключением того, что поле константы вместо одного значения представляет собой диапазон зна-

чений. Проверка на битовую эквивалентность проводится для каждого значения константы в диапазоне.

Сначала зададим адресную шину, как показано ниже:

```
FIELD address = [A3..A0]
```

Затем запишем уравнение с использованием диапазона значений:

```
select = address:[C..F] ;
```

Что эквивалентно следующему уравнению:

```
select = address:C # address:D # address:E # address:F;
```

Использование таблиц истинности

Иногда одним из самых понятных способов описаний функционирования устройства является использования таблиц данных. В языке CUPL ключевое слово **TABLE** указывает на то, что далее следует такая таблица. Вначале определим соответствующие списки для входных и выходных переменных. Затем установим однозначное соответствие между декодированными значениями списков входных и выходных переменных. Не следует обращать внимание на то, что значения входные значения находятся в левых частях выражений.

Список входных значений может быть определен таким образом чтобы одним оператором можно было задать много назначений. Блок операторов, приведенный ниже описывает простой преобразователь из шестнадцатеричного кода в десятичный.

```
FIELD input = [in3..0] ;  
FIELD output = [out4..0] ;  
TABLE input => output {  
0=>00; 1=>01; 2=>02; 3=>03;  
4=>04; 5=>05; 6=>06; 7=>07;  
8=>08; 9=>09; A=>10; B=>11;  
C=>12; D=>13; E=>14; F=>15;  
}
```

Конечные автоматы

Конечным автоматом называется устройство, которое упорядоченным способом проходит заданную последовательность состояний. Синхронный конечный автомат представляет собой логическую схему с триггерами. Так как его выход можно соединить с его собственным входом, а также входами некоторых других триггеров, то входное значение триггера может зависеть как от состояния его собственного выхода так и от состояний на выходах других триггеров. Следовательно конечное выходное состояние триггера зависит как от его предыдущего состояния, так и от состояний связанным с ним других триггеров.

Модель комбинационного автомата языка CUPL, показанная ниже, использует шесть компонентов: входная комбинаторная логическая схема, запоминающие регистры, биты состояния, выходы регистров, выходы логической схемы.

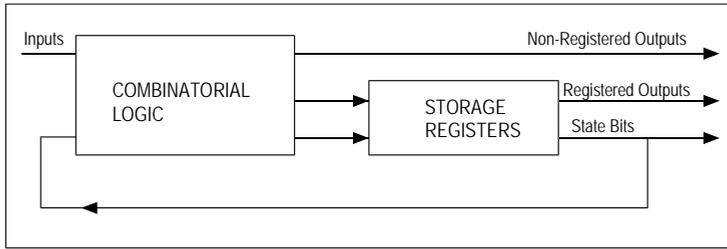


Рис. 5.7. Модель конечных автоматов

Inputs (входы)

Это входные сигналы устройства, генерируемые какими-либо другими устройствами.

Combinatorial Logic (комбинаторная логическая схема)

Любая комбинация логических вентилях (обычно **AND** и **OR**), которая порождает выходной сигнал, появляющийся спустя время задержки распространения **T_{pd}** при подаче любых сигналов с устройства управляющего этими вентилями. **T_{pd}** представляет собой время задержки между моментом возбуждения входа или цепей обратной связи и моментом появления сигнала на выходах комбинаторной логической схемы.

State bits (биты состояния)

Сигналы обратной связи с выходов регистров для управления комбинаторной логикой. Эти сигналы содержат информацию о предыдущем состоянии регистров.

Storage Registers (запоминающие регистры)

Запоминающие регистры — это триггеры любого вида, на входы которых подаются сигналы с комбинаторной логической схемы. Одни регистры используются для получения битов состояния, другие — для буферизации выходов. Сигналы на выходах регистров появляются через время **T_{co}** после подачи тактового импульса. **T_{co}** представляет собой время задержки между моментом возбуждения входа тактового сигнала и моментом появления сигнала на выходах триггеров.

Для успешной реализации конечных автоматов язык CUPL имеет синтаксис, позволяющий описать любую функцию внутри них. Ключевое слово **SEQUENCE** определяет выходы конечных автоматов, за которыми следуют операторы определяющие выполняемую ими функцию. Ключевое слово **SEQUENCE** определяет тип запоминающего регистра и тип выхода регистра как тип по умолчанию для конкретного целевого устройства. Кроме того существуют и другие ключевые слова, производные от **SEQUENCE**: **SEQUENCED**, **SEQUENCEJK**, **SEQUENCERS** и **SEQUENCET**. Они принудительно задают тип запоминающего регистра и тип выхода регистра как D-, JK-, SR и T-триггеры соответственно. Формат оператора **SEQUENCE** следующий:

```
SEQUENCE state_var_list {
    PRESENT state_n0
        IF (condition1) NEXT state_n1;
        IF (condition2) NEXT state_n2    OUT out_n0;
```

```
    DEFAULT    NEXT  state_n0;
    PRESENT    state_n1
    NEXT    state_n2;
    .
    .
    .
    PRESENT    state_nn statements ;
}
```

где

state_var_list — список переменных битов состояния, используемых в блоке конечных автоматов. Список переменных может быть представлен полем переменной.

state_n — индекс состояний, является дешифрованным значением **state_var_list** и должен быть уникальным для каждого оператора **PRESENT**.

statements — операторы условия, операторы **NEXT** или **OUT**, описываемые далее в следующих подразделах этого раздела.

Наборы конечных автоматов

Синтаксис языка CUPPL позволяет в одном проекте описать несколько конечных автоматов. Иногда разработчику хотелось бы соединить друг с другом несколько конечных автоматов таким образом, что после того, когда один автомат достигнет определенного состояния, начинает функционировать другой. Существует два способа выполнить такое соединение: использовать ряд операций над битовыми состояниями или задать глобальный регистр, доступ к которому могут иметь оба конечных автомата.

Условный оператор одного конечного автомата может содержать имя другого, за которым следует индекс состояния или диапазон индексов. Условный оператор примет значение ИСТИНА при достижении определенного состояния другим автоматом. Тот же результат будет достигнут при использовании регистра, доступного нескольким комбинационным автоматам. Однако для этого способа понадобится использование одного из выходов устройства или одного внутреннего регистра. В зависимости от ситуации, глобальный регистр также может быть комбинаторным, что может внести разнообразие в поведение конечного автомата в момент получения информации от другого конечного автомата.

Оператор условия

Оператор **CONDITION** обеспечивает определение логических функций на более высоком уровне, чем при их описании стандартными логическими выражениями булевой алгебры.

```
CONDITION {
    IF expr0 OUT var ;
    .
    .
    IF exprn OUT var ;
}
```

```

DEFAULT OUT var ;
}

```

Синтаксис оператора **CONDITION** эквивалентен таковому оператору асинхронного условного выхода конечного автомата, за исключением того, что там нет ссылки на конкретное состояние. Переменная логически декларируется всякий раз, когда встречается логическое выражение или условие **DEFAULT**.

Определение пользовательских функций

Ключевое слово **FUNCTION** позволяет создавать пользовательские ключевые слова посредством описания подразумеваемых под ними логических выражений и задания имени. Это имя затем может быть использовано в любых логических выражениях для вызова данной функции. Формат записи пользовательских функций следующий:

```

FUNCTION name ([parameter0, ..., parameterN])
{
    body
}

```

Операторы внутри тела функции могут определять высказывание функции или могут быть несвязанными уравнениями.

Число параметров как при определении, так и при вызове функции должно быть одинаково. Параметры определенные в теле функции заменяются параметрами при вызове функции, указанными в логических уравнениях. Переменная, используемая при вызове функции определяется телом функции. Если в операторах, представленных в теле функции не сделано ни каких определений, переменной при вызове функции переменной присваивается значение 'h'0.

Компиляция проекта на ПЛИС

В этом разделе рассказывается о том, как правильно выбрать целевую микросхему, настроить параметры компилятора проекта на ПЛИС и задать нужные выходные форматы.

Выбор целевой микросхемы

Компилятор проектов на базе ПЛИС, как и любой другой компилятор, обрабатывает исходный файл и генерирует выходной файл, основываясь на архитектуре указанного целевого устройства. Количество архитектур, поддерживаемых данным компилятором, превышает несколько сотен. Это позволяет не производить значительных изменений в исходном файле при реализации проекта на базе другой архитектуры.

Целевое устройство (конкретный тип микросхемы) выбирается в диалоговом окне **Configure PLD**. На вкладке **Options** отражается заданный на данный момент тип микросхемы. Изменение типа устройства производится щелчком мыши по кнопке **Change**, что вызовет появление диалогового окна **Target Device**.

При компиляции проекта **CUPPL** компилятором можно воспользоваться опцией виртуального устройства (**Virtual Device**). Она позволяет проверить, будет ли вообще ком-

пилироваться проект, а также определить требуемое необходимое максимальное количество комбинационных логических блоков (термов производства). Их число приводится в **DOC** файле. Большое число КЛБ в конечном счете может стать причиной использования кристалла с неоправданно большим их объемом. Чем больше блоков необходимо для проекта, тем больше их должно быть в кристалле. Различные комбинации опций и параметров оптимизации компилятора позволяют сократить число требуемых комбинационных логических блоков.

Задание типа микросхемы в файле источнике

Тип микросхемы, на котором будет реализован данный проект можно задать в заголовке исходного файла компилятора в поле **DEVICE**. При выборе другого типа микросхемы в диалоговом окне **Configure PLD** это поле автоматически обновляется.

Задание типа микросхемы в диалоговом окне Target Device

Диалоговое окно **Target Device** вызывается щелчком левой кнопки мыши по кнопке **Change**, расположенной в окне **Configure PLD**. В окне **Target Device** отображается список и краткое описание всех поддерживаемых типов микросхем (рис. 5.8). Чтобы изменить целевое устройство, необходимо выбрать ее тип и имя, после чего нажать кнопку **OK**.

Библиотеки устройств

Информация, отображаемая в диалоговом окне **Target Device**, содержится в библиотеках устройств. Для каждого типа микросхемы, показанного в поле **Device Type**, имеются библиотеки устройств в виде файлов с расширением **.DL**. В этих бинарных файлах содержится описание каждого поддерживаемого компилятором устройства. Библиотеки описывают физические характеристики каждого устройства, включая внутреннюю архитектуру, число выводов, действующие входные и выходные выводы. Также библиотеки характеризуют их логические характеристики: выводы регистров и комбинаторной логики, число термов производства, информацию карт прошивки и информацию о загружаемом формате.

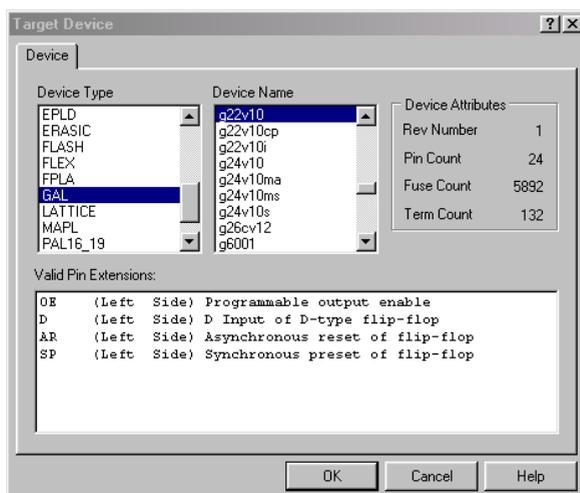


Рис. 5.8. Выбор целевой микросхемы.

Мнемоническое обозначение устройств

Система обозначений микросхем существует для того, чтобы идентифицировать различные архитектуры целевых микросхем. Очень важно заметить, что эта система обозначений предназначена только для архитектур, а не для конкретных микросхем. Например, микросхемы Altera EP512 и Intel 5AC312 оба имеют одинаковую архитектуру. Следовательно, если пользователь выполнял свой проект на одном из указанных кристаллов, то он должен был задать компилятору целевую архитектуру EP312.

Мнемоническая схема может указывать на множество производителей. Для того, чтобы узнать мнемоническую схему нужного устройства, необходимо его найти в файле **DEVHELP.TXT**. После чего в файле **DEVICES.TXT** определить, у какого производителя есть в списке микросхема, выбираемая в диалоговом окне **Target Devices**. Пользователь может напрямую впечатать мнемонику в поле целевого устройства **Target Devices** диалогового окна **Configure PLD**.

Мнемоническое обозначение состоит из префикса, определяющего серию микросхем, и суффикса, который является номером промышленного стандарта. Ниже в таблице 5.27 приведен список префиксов мнемонических обозначений микросхем.

Таблица 5.27

Префикс	Серия микросхем
EP	Стираемая программируемая логическая микросхема (Erasable Programmable Logic Device, EPLD)
G	Общая матричная логика (Generic Array Logic, GAL)
F	Логическая матрица с эксплуатационным программированием (Field Programmable Logic Array, FPLA)
F	Вентильная матрица с эксплуатационным программированием (Field Programmable Gate Array, FPGA)
F	Устройство с эксплуатационным программированием, задающее логическую последовательность (Field Programmable Logic Sequencer, FPLS)
F	Генератор последовательностей с эксплуатационным программированием (Field Programmable Sequence Generator, FPSG)
P	Программируемая логическая матрица (Programmable Logic Array, PAL)
P	Программируемое логическое устройство (Programmable Logic Device, PLD)
P	Электрически стираемая программируемая логика (Programmable Electrically Erasable Logic, PEEL)
PLD	Псевдологическое устройство (Pseudo Logic Device)
RA	Биполярное программируемое ПЗУ (Bipolar Programmable Read-Only Memory, PROM)

Например, мнемоническое обозначение микросхемы PAL10L8 — P10L8, а микросхемы 82S100 — F100. Для биполярного ПЗУ суффиксом является размер матрицы. Например, мнемоническое обозначение микросхемы биполярного ПЗУ емкостью 1024 x 8 — RA10P8, так как она имеет 10 адресных входов и 8 выходов данных.

Виртуальное устройство

Опция виртуального устройства позволяет пользователю создать проект цифрового устройства на программируемой логике, не привязываясь к конкретному типу целевой микросхемы. Виртуальное устройство не является устройством в физическом смысле. Просто для виртуального устройства снимаются ограничения компилятора на количество термов производства и выводов, а также на использование различных типов регистров. Опцию виртуального устройства полезно использовать для определения ресурсов, необходимых для реализации проекта.

Для виртуального устройства компилятор игнорирует полярность в описании выводов. Номера выводов также могут быть опущены.

Опции компилятора

Вкладка **Options** диалогового окна **Configure PLD Compiler** (рис. 5.9) дает пользователю возможность:

- Выбрать целевое устройство.
- Установить опции оптимизации.
- Установить опции компиляции.
- Установить уровень минимизации логики.
- Определить область действия идентификаторов цепей при компиляции для проектов на базе ПЛИС, описанных принципиальной схемой.

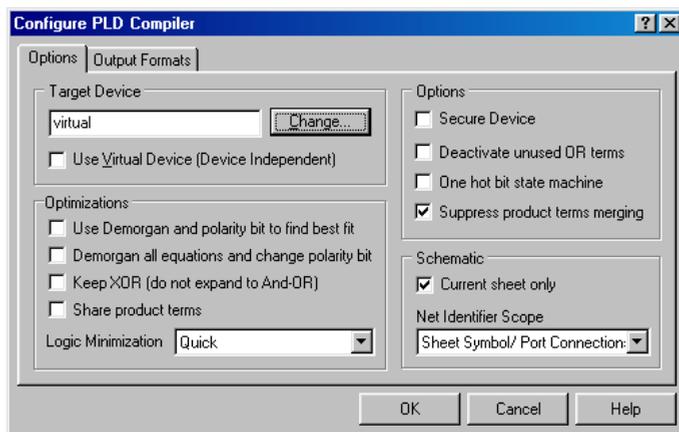


Рис. 5.9. Окно конфигурации компилятора.

Описание используемых опций приведено в таблице 5.28.

Таблица 5.28

Опция	Описание
Use DeMorgan and polarity bit	Оптимизирует терм произведения, использующий переменные выводов или узлов. Если в исходном файле встречается оператор DEMORGAN, то он обновляется.
DeMorgan all equations	Все переменные выводов или узлов подвергаются процессу ДеМоргановизации. Если в исходном файле встречается оператор DEMORGAN, то он обновляется.
Keep XOR	Уравнения XOR не разлагаются до уравнений вида AND-OR. Используется для независимых от типа кристалла проектов или проектов нацеленных на упаковываемые микросхемы, где модуль упаковки поддерживает XOR вентили..
Share Product Terms	Стимулирует разделение термов произведения во время минимизации и разрешает приведение групп.
Secure Device	Добавляет необходимый код в файл загрузки JEDEC формата для автоматического прожигания программатором бита безопасности при программировании, что делает микросхему не стираемой. Не все программаторы поддерживают эту опцию.
Deactivate unused OR terms	В устройствах структуры IFL матрица вентилях ИЛИ (OR) управляется отдельным вентилях И (AND). Обычно неиспользуемые входы вентилях ИЛИ оставляются компилятором подсоединенными к матрице термов произведения. В этом случае можно добавить новые элементарные ячейки. Однако при установке этой опции неиспользуемые входы вентилях ИЛИ удаляются из матрицы термов произведения. Как результат этого — уменьшение задержки распространения с входа на выход.

Таблица 5.28. Продолжение

Опция	Описание
Suppress product terms merging	В устройствах структуры IFL каждый терм производства из матрицы вентиляей И может совместно использоваться любым числом выходов вентиляей ИЛИИ. Эта опция отменяет данную возможность, стимулируя при необходимости создание идентичных термов производства для каждого из выходов матрицы ИЛИИ. Как результат этого — уменьшение задержки распространения с входа на выход. Эта опция форсирует процесс минимизации для каждого выхода в отдельности (в отличии от минимизации по всем выходам сразу), если выбран метод минимизации McCluskey или Espresso.
One-hot-bit State Machines	Эта опция предназначена для разработчиков аппаратуры на FPGA устройствах. Она принуждает компилятор формировать уравнения конечных автоматов как one-hot-bit. Это дает некоторые особые преимущества для устройств с насыщенной регистрами архитектурой, таких как микросхемы фирмы XILINX. При этом уменьшается веерность проекта, существенно облегчая трассировку и устраняя временные проблемы, связанные с изменяющимися длинами обратных связей от регистра к регистру. Если используется эта опция, компилятор легко обрабатывает все конечные автоматы, как one-hot-bit.
Net Identifier Scope	Определяет, как создаются связи между листами в многолистовом проекте. Существует два эффективных способа создания соединений: по горизонтали (глобальные метки цепей и порты и глобальные только порты), где соединения создаются непосредственно между подчиненными листами; и по вертикали (соединения типа символ листа — порт), где связи создаются от подчиненного листа к соответствующему входу листа.
Current Sheet Only	Используется, если необходимо откомпилировать только текущий лист принципиальной схемы.

Методы минимизации

Методами минимизации называются используемые компилятором специальные алгоритмы уменьшения избыточности логической схемы. В распоряжении пользователя системы Protel 99 SE имеется четыре метода минимизации:

- **Quick**
- **Quine-McCluskey**
- **Presto**
- **Espresso**

Уровни минимизации **Quine-McCluskey** и **Presto** выполняют многократную минимизацию выходов для устройств структуры **IFL**. Они обеспечивают максимум совместного использования термов произведения для устройств этой структуры.

Выбор слова **None** в окне метода минимизации запрещает компилятору проводить минимизацию логики. Это полезно, например, при работе с ПЗУ для сохранения от уничтожения содержащихся в нем термов произведения.

На рисунке 5.10 справа приведены сравнительные характеристики по эффективности, использованию памяти и скорости этих четырех методов минимизации.

Булева алгебра

В таблице 5.29 приведены правила Булевой алгебры для сокращения избыточных термов произведения из расшифрованных уравнений. Эти правила используются алгоритмами уменьшения избыточности логической схемы, встроенными в компилятор.

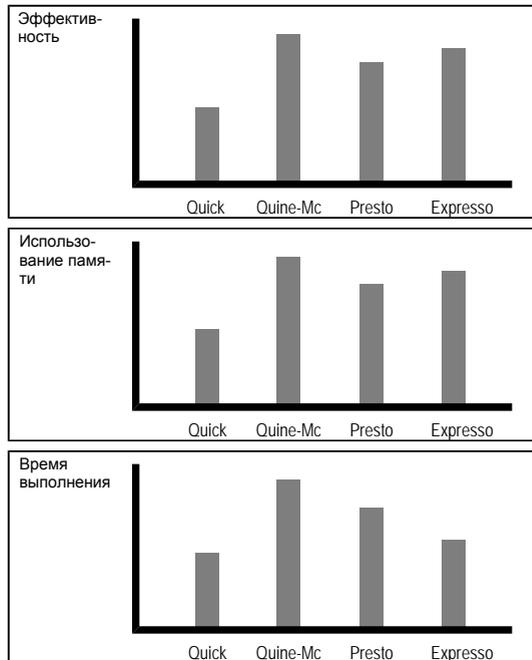


Рис. 5.10. Сравнение различных методов минимизации

Таблица 5.29

Логическое выражение	Результат
$\!0$	1
$\!1$	0
$A \& 0$	0
$A \& 1$	A
$A \& A$	A
$A \& \!A$	0
$A \# 0$	A
$A \# 1$	1
$A \# A$	A
$A \# \!A$	1
$A \& (A \# B)$	A
$A \# (A \& B)$	A

Форматы выходных файлов

Компилятор проектов на базе ПЛИС может создавать следующие типы выходных файлов.

- Загружаемые форматы — форматы, готовые для загрузки в программатор.
- Выходные форматы, включающие форматы интерфейса с упаковщиками третьих фирм и листинги ошибок.
- Форматы документации, которые могут содержать схему прошивок и уравнения.
- Формат моделирования, который по существу является программируемым виртуальным устройством. Программа моделирования подключает тестовые векторы из файла описания тестовых спецификаций с расширением **.SI** к этому виртуальному устройству, а результаты записывает в файл листинга программы моделирования **.SO**.

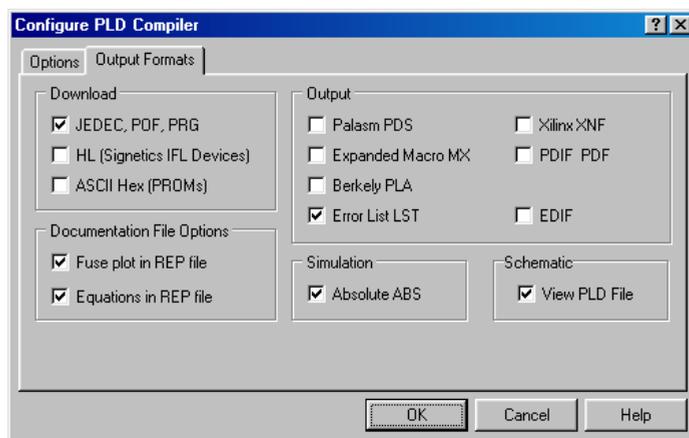


Рис. 5.11. Настройка выходного формата компилятора.

Конфигурирование выходных форматов компилятора заключается во включении соответствующих опций на вкладке **Output Formats** диалогового окна **Configure PLD**.

Описание каждого из выходных форматов приведено в таблице 5.30.

Таблица 5.30

Выходные форматы	Описание формата
JEDEC, POF, PRG	Создает JEDEC-совместимый файл загрузки в ASCII коде с расширением .JED. Имя файла определяется оператором NAME в секции заголовка файла описания логической схемы.
HL	Создает HL-файл загрузки с расширением .HL. Этот формат доступен только для приборов со структурой IFL фирмы Segnetic. Имя файла определяется оператором NAME в секции заголовка файла описания логической схемы.
ASCII Hex	Создает Hex-файл загрузки в формате ASCII с расширением .HEX. Этот формат подходит только для устройств ПЗУ (PROM). Имя файла определяется оператором NAME в секции заголовка файла описания логической схемы.
PALASM PDS	Создает файл формата PALASM с расширением .PDS для стандарта, установленного компанией Monolithic Memories, описанного в третьем издании справочника по устройствам PAL (PAL Handbook, Third Edition).
Expanded Macro MX	Создает файл расширенных макроопределений с расширением .MX, содержащий развернутый листинг всех используемых макросов в файле источнике. Кроме того, генерируемый файл содержит развернутые выражения, используемые командой REPEAT.
Berkeley PLA	Создает файл формата Berkeley PLA с расширением .PLA, для его использования инструментариями Berkeley PLA типа PLEASURE или другими вспомогательными программными средствами компоновки PLA, использующими формат Berkeley PLA. Этот формат применяется как входной для ряда внешних программ упаковки программируемой логики.
Error list LST	Создает файл листинга ошибок с расширением .LST. В исходном файле все строки пронумерованы. Сообщения об ошибках, перечисленные в конце файла, используют эти номера для ссылок.
XNF Xilinx	Создается файл, который может быть применен для создания входного файла других сервисных программ проектирования логических устройств и программ упаковки, таких как PDS2XNF компании Xilinx.

Таблица 5.30. Продолжение

Выходные форматы	Описание формата
PDIF PDF	Создает файл формата PDIF (P-CAD Database Interchange Format) с расширением .PDF, который может быть считан программой PDIFIN в принципиальную схему для обработки в схемном редакторе PC-CAPS (P-CAD Schematic Capture). Генерируемые символы будут содержать упаковочную информацию для ПЛИС.
EDIF	Создает файл формата EDIF.
Equations in DOC File	Создает файл документации с расширением .DOC, содержащий расширенный список логических элементов в формате суммы термов произведения и таблицу символов всех переменных, используемых исходным файлом. Он включает в себя общее количество термов произведения и число термов, доступных каждому отдельному выходу.
Fuse Plot in DOC File	Создает файл документации с картой прошивки. Для PAL-микросхем здесь перечисляется каждый выходной вывод, а ассоциированные с ним строки термов произведения показаны вместе с начальным номером JEDEC перемычек. Присутствующие перемычки обозначаются буквой X. Пережигаемые перемычки обозначаются знаком —. Для IFL устройств, использующих формат загрузки HL, номера JEDEC перемычек показываются при помощи входных термов, обозначаемых как H, L, 0, или —.
Absolute ABS	Создает файл с расширением .ABS, используемый программой моделирования ПЛИС
View PLD File	Автоматически открывает промежуточный PLD-файл, сгенерированный во время компиляции проекта на базе ПЛИС описанной принципиальной схемой.

Моделирование проекта на базе ПЛИС

В данном разделе показано, как создавать исходный файл тестовых спецификаций для моделирования проекта, основанного на применении программируемых логических интегральных схем (ПЛИС). Тестовые векторы определяют ожидаемое функционирование ПЛИС посредством определения выходных значений, как функций входных значений. Тестовые векторы используются и в процессе моделирования логической схемы, который производится перед программированием ПЛИС, и для функционального тестирования микросхемы ПЛИС после ее программирования. Программа моделирования ПЛИС способна генерировать совместимые с JEDEC форматом загружаемые тестовые векторы, которые добавляются в **.JED** файл, созданный при компиляции.

Входные данные для программы моделирования

Входными данными для программы моделирования являются данные из исходного файла тестовых спецификаций **.SI**. Он содержит функциональное описание требований, которым должна удовлетворять ПЛИС в схеме.

Задающие сигналы на входных выводах и тестовые значения на выходных выводах, введенные в исходном файле **.SI**, сравниваются с действительными значениями, вычисленными с помощью логических выражений, определенных в исходном файле проекта на базе ПЛИС с расширением **.PLD**. Эти вычисленные значения помещаются в файл абсолютных значений с расширением **.ABS**, который создается в процессе компиляции, если это задано в настройках. Такой файл должен быть создан во время компиляции до запуска программы моделирования.

Результаты моделирования

Результатами моделирования являются:

- листинг программы моделирования.
- векторы, добавляемые в загружаемый файл прошивки ПЛИС формата JEDEC.

Выходной файл-листинг с расширением **.SO** содержит результаты процесса моделирования, его имя совпадает с именем файла тестовых спецификаций.

Файл-листинг имеет обычный ASCII формат. Все данные из заголовка переносятся в файл-листинг с добавлением соответственным образом помеченных ошибок в этом заголовке. Каждому из векторов присваивается индивидуальный номер. Все закончившиеся неудачей выходные тесты помечаются фактическим (определяемым программой моделирования) выходным значением. Все переменные из ошибочного выражения помещаются в список вместе с ожидаемыми (заданными пользователем) значениями. Все неправильные или не совпавшие с ожидаемыми тестовые значения помещаются в список вместе с соответствующими сообщениями об ошибках.

Результаты моделирования могут быть показаны на экране в виде набора временных диаграмм сигналов. Для этого необходимо сделать двойной щелчок мыши на файле с расширением **.SO**, после чего он откроется редактором сигналов ПЛИС (PLD Waveform Editor).

Программа моделирования добавляет тестовые векторы в существующий загружаемый файл прошивки ПЛИС формата JEDEC (файл с расширением **.JED**), созданный во время компиляции. Для его создания необходимо включить опцию JEDEC во вкладке **Formats**, расположенной в диалоговом окне **Configure PLD**.

Создание исходного файла тестовых спецификаций

Файл тестовых спецификаций **.SI** имеет ASCII формат и создается с помощью любого текстового редактора. Его имя за исключением расширения совпадает с именем соответствующего исходного файла описания логической схемы на языке CUPL. Файл тестовых спецификаций должен содержать следующую информацию:

- Заголовок.
- Комментарии.
- Порядок переменных.
- Основания систем исчисления.
- Тестовые векторы.
- Директивы программе моделирования.

Заголовок

Информация в заголовке должна быть идентична информации в соответствующем файле описания логической схемы на языке CUPL (**.PLD**). При не совпадении данных выдается предупреждающее сообщение о том, что статус логических выражений может противоречить текущим тестовым векторам, находящимся в файле тестовых спецификаций.

Ниже перечислены ключевые слова, используемые в заголовке:

```
PARTNO  
NAME  
REVISION  
DATE  
DESIGNER  
COMPANY  
ASSEMBLY  
LOCATION  
DEVICE  
FORMAT
```

Наиболее простой способ создания файла тестовых спецификаций — копирование заголовочной информации из соответствующего исходного CUPL файла.

Определение микросхемы

Программа моделирования берет информацию о той или иной микросхеме из специальных библиотек с расширением **.DL**. Библиотеки описывают физические характеристики каждой микросхемы, включая внутреннюю архитектуру, количество выводов, тип доступных регистров, логические характеристики (включая наличие буферизованных и небуферизованных выводов), возможность обратных связей, состояние регистров после подачи питания и особенности регистрового управления.

Ссылки на целевые микросхемы производится с помощью мнемонических обозначений. Каждое такое обозначение состоит из префикса, определяющего серию микросхем, и суффикса, который является номером промышленного стандарта. Список префиксов мнемонических обозначений микросхем приведен в таблице 5.27.

Например, мнемоническое обозначение микросхемы PAL10L8 — P10L8, а микросхемы 82S100 — F100. Для биполярного ППЗУ суффиксом является размер матрицы. Например, мнемоническое обозначение микросхемы биполярного ППЗУ емкостью 1024 x 8 — RA10P8, так как она имеет 10 адресных входов и 8 выходов данных.

Целевая микросхема может быть указана двумя способами — либо в файле тестовых спецификаций **.SI**, либо во вкладке **Options** диалогового окна **Configure PLD**.

Комментарии

Комментарии используются для пояснения содержимого файла и назначения того или иного тестового вектора. Находиться они могут в любом месте файла тестовых спецификаций. Комментарий всегда начинается с набора символов (*****) и заканчивается символами (***/**). Он может занимать несколько строк, но не может быть вложенным.

Операторы

Программа моделирования для записи операторов в исходном файле предусматривает применение ключевых слов **ORDER**, **BASE** и **VECTORS**, которые определяют результаты моделирования и то, как они отображаются. Приведенные ниже разделы описывают форму записи операторов с помощью ключевых слов языка CUPL.

Оператор ORDER

Ключевое слово **ORDER** применяется для составления списка переменных, которые будут использованы в таблице моделирования, а также для определения того, как эти переменные будут отображаться. Обычно имена переменных совпадают с именами из соответствующего CUPL файла описания логической схемы. Форма записи такова: после слова **ORDER** ставится двоеточие, все переменные в списке разделяются запятой, список завершается точкой с запятой. Далее приведены примеры этого оператора:

```
ORDER: inputA, inputB, output ;
```

В списке должны быть перечислены только те переменные, которые действительно используются в процессе моделирования.

Полярность имен переменных может отличаться от той, которая была объявлена в CUPL файле описания логической схемы, что позволяет моделировать выходы с активным низким уровнем с помощью векторов с активным высоким уровнем. Имена переменных могут следовать в любом порядке, так как программа моделирования автоматически создает надлежащий порядок и подбирает полярности элементов результирующего вектора так, чтобы это соответствовало требованиям JEDEC формата для данной микросхемы. Если в операторе **ORDER** используются индексные переменные, они могут быть выражены в формате условных знаков списка (list notation format). Однако, так как оператор **ORDER** сам по себе представлен в форме списка, пропадает необходимость использования квадратных скобок при определении диапазона массива переменных. Далее приведен пример двух эквивалентных друг другу операторов, но первый представлен перечислением всех переменных, а второй — в форме списка:

```
ORDER: A0, A1, A2, A3, SELECT, !OUT0, !OUT1;  
ORDER: A0..3, SELECT, !OUT0..1 ;
```

В примере, где оператор представлен в форме списка, полярность первой индексной переменной (**!OUT0**) определяет полярность всего списка. Ссылки на битовые поля, объявленные в CUPL файле описания логики, могут быть сделаны посредством имен их одиночных переменных. Битовые поля также могут быть объявлены в файле тестовых спецификаций, который предназначен для программы моделирования, с помощью использования ключевого слова **FIELD**. Объявление оператора **FIELD** должно предшествовать оператору **ORDER**.

Оператор **ORDER** может использоваться для определения формата векторных результатов программы моделирования, которые помещаются в файл списка. По умолчанию значения переменных записываются без пробелов между столбцами. Например, приведенный ниже оператор **ORDER**

```
ORDER: clock, input, output ;
```

генерирует следующую запись в выходной файл:

```
0001: C0H  
0002: C1L
```

Пробелы между столбцами могут быть добавлены с помощью символа (%) и десятичного значения от 1 до 80. Например, приведенный ниже оператор **ORDER**

```
ORDER: clock, %2, input, %2, output ;
```

генерирует следующую запись в выходной файл:

```
0001: C 0 H  
0002: C 1 L
```

Заметим, что оператор **ORDER** должен заканчиваться точкой с запятой.

Текст в выходной файл можно вставить, заключив текстовую строку в двойные кавычки. Например, приведенный ниже оператор **ORDER**

```
ORDER: "Clock is ", clock,  
      " and input is ", input,  
      " output goes ", output ;
```

в выходном файле будет присутствовать:

```
0001: Clock is C and input is 0 output goes H  
0002: Clock is C and input is 1 output goes L
```

Использование нескольких операторов ORDER

В одном **.SI** файле могут присутствовать несколько операторов **ORDER**. Например, если файл **TEST.SI** будет содержать следующий текст:

```
Name      test;  
Partno    XXXXX;  
Date      XX/XX/XX;  
Revision  XX;  
Designer  XXXXX;  
Company   XXXXX;  
Assembly  XXXXX;  
Location  XXXXX;  
Device    g16v8;  
  
Order: A, %1, B, %1, X, %1, Y;  
Vectors:  
  0 0 H L  
  0 1 H H  
  1 0 H H  
  1 1 L L  
  0 X H X  
  X 0 H X  
  1 X X X  
  X 1 X X  
Order: A, B, X;  
Vectors:  
  0 0 H  
  0 1 H  
  1 0 H  
  1 1 L  
  0 X H  
  X 0 H  
  1 X X  
  X 1 X
```

содержимое файла **TEST.SO** будет выглядеть следующим образом:

```
CSIM: CUPL Simulation Program
Version 4.2a Serial# ...
Copyright (c) 1996 Protel International
CREATED Wed Dec 04 02:14:12 1991
LISTING FOR SIMULATION FILE: test.si
1: Name          test;
2: Partno        XXXXX;
3: Date          XX/XX/XX;
4: Revision      XX;
5: Designer      XXXXX;
6: Company       XXXXX;
7: Assembly     XXXXX;
8: Location      XXXXX;
9: Device        g16v8;
10:
11: Order: A, %1, B, %1, X, %1, Y;
12:
=====
      A B X Y
=====
0001: 0 0 H L
0002: 0 1 H H
0003: 1 0 H H
0004: 1 1 L L
0005: 0 X H X
0006: X 0 H X
0007: 1 X X X
0008: X 1 X X
25: Order: A, B, X; 26:
=====
      ABX
=====
0010: 00H
0011: 01H
0012: 10H
0013: 11L
0014: 0XH
0015: X0H
0016: 1XX
0017: X1X
```

Оператор **BASE**

В большинстве случаев каждая переменная в операторе **ORDER** (исключая переменные **FIELD**) имеют соответствующее тестовое значение, состоящее из одного символа, которое входит в таблицу тестовых векторов выходного файла. Если тестовый вектор

содержит несколько значений, то он может быть представлен в форме числа в кавычках. Для входных значений используются одинарные кавычки, для выходных — двойные. Оператор **BASE** определяет, каким образом раскрывается число в кавычках, то есть основание системы счисления. Формат этого оператора:

```
BASE: name;
```

где параметр `name` принимает значения **octal** (восьмеричное), **decimal** (десятичное) или **hex** (шестнадцатеричное).

Заметим, что за ключевым словом **BASE** следует двоеточие, а заключительным символом данного оператора является точка с запятой.

Шестнадцатеричное представление — это значение по умолчанию. Оператор **BASE** следует применять до оператора **ORDER**.

Если определена десятичная или шестнадцатеричная система счисления, число в кавычках будет разложено на четыре цифры, при восьмеричной — на три. Например, тестовый вектор '7' интерпретируется следующим образом:

```
1 1 1           основание восьмеричное
```

или

```
0 1 1 1       основание десятичное
```

или

```
0 1 1 1       основание шестнадцатеричное
```

В кавычках может быть введено несколько цифр. Например, запись '563' интерпретируется так:

```
1 0 1 1 1 0 0 1 1   основание восьмеричное
```

или

```
0 1 0 1 0 1 1 0 0 0 1 1   основание десятичное
```

или

```
0 1 0 1 0 1 1 0 0 0 1 1   основание шестнадцатеричное
```

Значения в кавычках также могут быть использованы с другими тестовыми обозначениями. Например, при восьмеричном основании

```
"XX"           раскладывается на X X X X X X
```

```
"LL"           раскладывается на L L L L L L
```

```
"45"           раскладывается на H L L H L H
```

Заметим, что значения в кавычках не могут содержать символа (*).

Тестовые значения для переменных в операторе **FIELD**, который объявляет массив переменных, могут быть выражены непосредственно (например, 001, HHLL) или значением в кавычках (например, '1', "C"). Когда используется форма представления с кавычками, значение автоматически раскладывается на количество переменных в данном массиве. Например, следующий адресный массив

```
FIELD address = [A0..5] ;
```

имеющий тестовое значение

```

/*
A      A      A      A      A
5      4      3      2      1      0
-----*/
1      1      1      0      0      1
    
```

может быть записан либо как единичные тестовые значения, либо как значения в кавычках — '39'.

Оператор VECTORS

Ключевое слово **VECTORS** используется в качестве префикса таблицы тестовых векторов. За ключевым словом следуют тестовые векторы, записанные либо как единичные тестовые значения, либо как значения в кавычках. Каждый вектор должен располагаться на отдельной строке, в конце которой не должно быть символа точки с запятой. В приведенной ниже таблице перечислены допустимые значения тестовых векторов:

Таблица 5.31

Значение	Описание
0	Вход, управляемый низким уровнем (0 вольт)
1	Вход, управляемый высоким уровнем (+5 вольт)
C	Тактовый вход, управляемый фронтом (0, 1, 0)
K	Тактовый вход, управляемый спадом (1, 0, 1)
L	Тестовый выход имеет напряжение низкого уровня (0 вольт)
H	Тестовый выход имеет напряжение высокого уровня (+5 вольт)
Z	Тестовый выход в высокоомном (третьем) состоянии
X	Вход имеет напряжение высокого или низкого уровня, выход имеет напряжение высокого или низкого уровня. Заметим, что не все программаторы трактуют X на входе одинаково — некоторые устанавливают вход в 0, другие позволяют этот вход присоединять к 1 с помощью резистора, остальные оставляют состояние входа в предыдущем значении.
N	Выход не тестируется
P	Предварительная загрузка внутреннего регистра (значение появится на выходе !Q)
R	Генератор случайного входного значения
*	Только выходы. Программа моделирования определяет тестовое значение и подставляет его в вектор
"	Символы, в которые заключается значение входа, интерпретирующееся согласно оператору BASE (octal, decimal или hex). Допустимые значения: 0..F и X.
""	Символы, в которые заключается значение выхода, интерпретирующееся согласно оператору BASE (octal, decimal или hex). Допустимые значения: 0..F, H, L, Z и X.

Пример таблицы тестовых векторов:

```
VECTORS:
0 0 1 1 1 'F' Z "H" /* test outputs HI */
0 1 1 0 0 '0' Z "L" /* test outputs LO */
```

Генерация случайного входного значения

При использовании значения **R** для соответствующего сигнала в тестовом векторе генерируется случайное значение (0 или 1).

Пример из **.SI** файла:

```
$repeat 10;
C 0 RRR 1RRRRRRR *****
```

приведет к следующему результату в **.SO** файле:

```
0035: C 0 000 10001011 HLLLHLHH
0036: C 0 000 11100111 HHHLLHHH
0037: C 0 110 10111101 HHHHLHHL
0038: C 0 111 11000100 HLLLHLLH
0039: C 0 101 10001011 LHLHHHLL
0040: C 0 101 10000110 LLHHLHLL
0041: C 0 010 10000001 LHHLLLLL
0042: C 0 000 10010000 HLLHLLLL
0043: C 0 001 11110100 LHHHHLHL
0044: C 0 001 10011110 LHLLHHHH
```

NOT : ones complement !		AND &		
A	!A	A	B	A & B
0	1	0	0	L
1	0	0	1	L
X	X	0	X	L
		1	0	L
		1	1	H
		1	X	X
		X	X	X

OR #			XOR : exclusive OR \$		
A	B	A # B	A	B	A \$ B
0	0	L	0	0	L
0	1	H	0	1	H
0	X	X	0	X	X
1	0	H	1	0	H
1	1	H	1	1	L
1	X	H	1	X	X
X	X	X	X	X	X

Рис. 5.12. Таблица истинности векторов

Состояние X

Программа моделирования ПЛИС, в отличие от многих других подобных программ, трактует состояние X, как любое другое значение, а не полагает, что X для входа — это 0, а для выхода — N. Состояние X позволяет определить, какие входные значения влияют на состояние данного выхода, согласно правилам из таблиц истинности, пример которых приведен ниже.

Предварительная загрузка

Тестовое значение P используется на тактовом выводе буферизованного устройства для предварительной загрузки нужным значением внутренних регистров проекта конечного автомата или счетчика в том случае, когда это устройство не имеет специально предназначенного для такой загрузки ТТЛ вывода. Для действительной загрузки регистров программатор использует повышенное напряжение. Все входные выходы микросхемы ПЛИС игнорируются и, поэтому, должны быть определены как X. Значения, определенные в регистровых переменных, загружаются на выходы !Q регистра. Эти значения (0 или 1) — абсолютные уровни независимо от полярности выходов и инвертирующих буферов. Далее приведен пример последовательности предварительной загрузки для переменной, определяющей выход с активным низким уровнем. Микросхема имеет инвертирующие буферы между выходом Q регистра и выводом самой микросхемы.

```
ORDER: clock, input1, input2 , !output ;
VECTORS:
P X X 1 /* reset flip-flop */
        /* !Q goes to 1 */
        /* Q goes to 0 */
0 X X H /* output is HI due to */
        /* inverting buffer */
```

Программа моделирования может моделировать и генерировать тестовые векторы предварительной загрузки даже для микросхем, которые не имеют возможностей предварительной загрузки. Однако, не все ПЛИС способны осуществлять загрузку с использованием повышенного напряжения. Такие микросхемы имеют специально предназначенные для этих целей выходы. Программа моделирования не проверяет, имеется ли возможность предварительной загрузки, так как микросхемы разных производителей имеют различные характеристики. Перед использованием функции предварительной загрузки проверяется лишь способность тестируемой микросхемы использовать для этих целей повышенное напряжение.

Тактовый сигнал

Большинство синхронных микросхем (то есть микросхем, которые содержат регистры с общим тактовым сигналом, связанным с некоторым выходным выводом) используют тактовый сигнал с активным передним фронтом. Для того, чтобы гарантировать правильную работу программы моделирования с такими микросхемами, необходимо на тактовом выводе всегда использовать тестовое значение C (не 1 или 0). Для синхронных микросхем с активным задним фронтом — тестовое значение K.

Асинхронные векторы

При записи тестовых векторов для схемы с асинхронными обратными связями одновременное изменение двух тестовых значений может создать условия для появления нежелательных выбросов, которые приведут к неправильным результатам. Схема, приведенная ниже, имеет три входа (А, В и С) и выход Y с обратной связью.

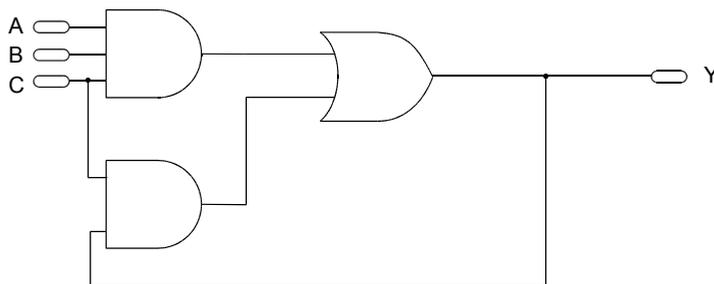


Рис. 5.13. Схема с обратной связью

Выражение для выхода Y:

$$Y = A \& B \& C \# C \& Y$$

Векторная таблица показывает, при каких входных значениях на выходе получается напряжение низкого уровня (рис. 5.14).

Так как по крайней мере один из входов в каждом из векторов равен 0, вентиль И со входами А, В и С имеет напряжение низкого уровня на выходе. Это напряжение низкого уровня на выходе Y, поступающее по цепи обратной связи, удерживает напряжение низкого уровня также и на выходе другого вентиля И. Таким образом, вентиль ИЛИ (управляемый выходами двух вентилях И), а, следовательно, и выход Y остаются в состоянии логического нуля для указанных в таблице векторов.

Однако, когда программатор оперирует тестовыми векторами, он применяет значения последовательно, начиная с первого вывода. Поэтому при изменении двух тестовых значений в соседних векторах программатор создает промежуточный результат (помеченный в таблице на рисунке 5.15 символом "а").

	A	B	C	Y
0001	0	0	0	L
0002	0	1	1	L
0003	1	0	1	L

Рис. 5.14. Векторная таблица для схемы с обратной связью

	A	B	C	Y
0001	0	0	0	L
0001a	0	1	0	L
0002	0	1	1	L
0002a	1	1	1	H
0003	1	0	1	H

Рис. 5.15. Векторная таблица с промежуточными результатами

Промежуточный результат [0002a] вырабатывает высокий уровень на выходе Y. Этот высокий уровень по цепи обратной связи попадает на вход вентиля И, на вход С которого в это время приходит 1, полученная из вектора [0003]. Поэтому на выходе вентиля И, и далее на выходе Y вентиля ИЛИ появляется высокий уровень, что противоречит ожидаемому значению, указанному в третьем векторе, то есть образовались условия для появления нежелательных выбросов.

Поэтому во избежание появления выбросов необходимо следить за тем, чтобы соседние векторы отличались только в одном значении. Для этих целей существует еще одна полезная функция — установка в исходном файле спецификаций параметра **TRACE** в значение 1, 2 или 3, тогда как по умолчанию он равен 0. Это заставит программу моделирования записать промежуточные результаты в выходной файл. Подробнее использование директивы **TRACE** описывается в подразделе *Директивы программы моделирования*.

Моделирование I/O выводов

При записи тестовых векторов для проекта, который имеет возможности ввода/вывода (I/O) и управляемое разрешение выхода (OE), значение тестового вектора для I/O выводов будет зависеть от значения OE. Если OE имеет низкий уровень, то I/O выводу необходимо выходное тестовое значение (L, H, *, ...). Как только сигнал OE становится неактивным, на I/O выводе появляется Z состояние. В этот момент на I/O вывод могут быть поданы входные тестовые значения (0, 1, ...), что позволит этому выводу стать входным. Когда сигнал OE станет снова активным, тестовые значения для данного вывода будут отражать выходы макроячейки.

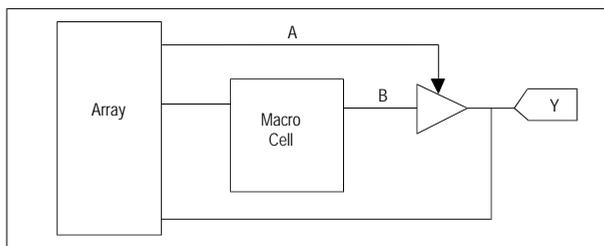


Рис. 5.16. Моделирование I/O вывода.

Булево выражение для данного рисунка выглядит следующим образом:

$$Y = B;$$

$$Y.OE = A;$$

Когда A равно значению ИСТИНА (**TRUE**), выход макроячейки B появляется на выводе Y. Когда A равно значению ЛОЖЬ (**FALSE**), сигнал разрешения выхода (OE) становится неактивным, и на выводе Y появится Z-состояние. Теперь на данный вывод можно подавать входные значения. Ниже приведен пример моделирующего файла:

```
Order: A, %1, B, %3, Y;
```

```
Vectors:
```

```
1 0 L /* OE is ON */
```

```
1 1 H
```

```
0 0 Z /* OE is OFF */
```

```
0 0 1 /* a valid input value can be placed on pin Y */
```

```
1 0 L /* OE is ON again */
```

Объявление переменных (VAR)

Синтаксис: VAR <var_name> = <var_list>;

<var_name> — строка длиной до 20 символов, которая может содержать буквы, цифры и символы подчеркивания, но не может заканчиваться цифрой.

<var_list> — список символов из оператора ORDER (одиночных, групповых или массивов), ранее определенных переменных; разделенных запятыми.

<var_list> = [!]<field> | [!]<group> | [!]<var> [..[!]<var> | ,<var_list>]

Действие:

С помощью такого объявления все объекты, входящие в список <var_list>, объединяются под общим именем для удобства при дальнейшем использовании. Это подобно оператору **FIELD** за исключением того, что оператор **VAR** не может быть применен до оператора **ORDER**, а должен располагаться между операторами **ORDER** и **VECTORS**.

Пример:

```
VAR Z = Q7..4;
```

Директивы программы моделирования

Программа моделирования имеет шесть директив, которые могут быть расположены в любой строке файла после оператора **VECTORS**. Имена всех директив начинается с символа \$, а сама директивная команда должна заканчиваться точкой с запятой. В число директив программ моделирования:

\$MSG \$SIMOFF

\$REPEAT \$SIMON

\$TRACE \$EXIT

\$MSG

Директива **\$MSG** используется для размещения в выходном файле программы моделирования документирующих сообщений или информации о форматировании. Например можно создать заголовок для таблицы функции или список имен переменных. Формат директивы такой:

```
$MSG "any text string" ;
```

В выходной таблице текстовая строка будет без кавычек.

В выходной файл можно вставить пустую строку, например, между тестовыми векторами, используя следующий формат:

```
$MSG "" ;
```

\$REPEAT

Директива **\$REPEAT** ведет к тому, что вектор будет продублирован указанное количество раз. Формат этой директивы:

```
$REPEAT n ;
```

где *n* — это десятичное число от 1 до 9999.

Следующий за директивой **\$REPEAT** вектор дублируется указанное число раз.

Директива **\$REPEAT** особенно полезна для тестовых счетчиков и смены состояний. Для представления выходных тестовых значений, вычисленных программой моделирования, используется символ (*). Приведенный ниже пример демонстрирует двухразрядный счетчик из CUPL файла и оператор **VECTORS**, в котором использована директива **\$REPEAT** для тестирования этого счетчика.

Из компилятора:

```
Q0.d = !Q0 ;
Q1.d = !Q1 & Q0 # Q1 & !Q0 ;
```

В программу моделирования:

```
ORDER: clock, input, Q1, Q0 ;
VECTORS:
0 0 X X /* power-on condition */
P X 1 1 /* reset the flip-flops */
0 0 H H
$REPEAT 4 ; /* clock 4 times */
C 0 * *
```

Такой файл генерирует следующие тестовые векторы:

```
0 0 X X
P X 1 1
0 0 H H
C 0 L L
C 0 L H
C 0 H L
C 0 H H
```

Далее программа моделирования работает с этим набором из четырех векторов.

\$TRACE

Директива **\$TRACE** используется для определения объема информации, относящейся к тестовым векторам, которую программа моделирования напечатает в процессе моделирования. Формат директивы **\$TRACE**:

```
$TRACE n ;
```

где n — десятичное число от 0 до 4, задает уровень трассировки.

Уровень 0 (устанавливается по умолчанию) выключает всю дополнительную информацию, печатаются только результирующие тестовые векторы. Когда в схеме используются небуферизованные обратные связи, значение сигнала, вычисленное для первого прохода (вектора), в цепи обратной связи неизвестно. Если новое значение в этой цепи изменяет значение на каком-либо выходе, тогда вектор вычисляется снова. Критерием того, что схема перешла в стабильное состояние, считается случай, когда состояние выходов не изменяется после двух проходов.

Уровень 1 печатает промежуточные результаты для всех векторов, которые требуют более одного прохода вычислений для установления стабильного состояния. Те векторы, которые требуют более двадцати вычислительных проходов рассматриваются как нестабильные.

Уровень 2 для проекта, использующего буферы, определяет три фазы моделирования. Первая фаза — "до тактового импульса", в которой вычисляются промежуточные векторы, использующие небуферизованные обратные связи. Вторая фаза — "во время тактового импульса", в которой содержимое регистров выдается сразу после прихода тактового импульса. Третья фаза — "после тактового импульса", в которой вычисляются значения на выходах, использующих обратные связи, как на уровне 1.

Уровень 3 обеспечивает самую высокую степень выдачи информации из программы моделирования. Печатаются все три фазы процесса моделирования, перечисленные на уровне 2, а также список индивидуальных для каждой переменной конъюнктивных выражений. Выходные значения вентиля И помещаются в список вместе с значениями входов данного вентиля.

Уровень 4 обеспечивает возможность отслеживать логическое значение до выходного буфера. При применении директивы **\$TRACE 4** программа моделирования сообщает только об значениях настоящих выходных выводов и присваивает значение ? входам и скрытым узлам. Для комбинаторных выходов уровень 4 выводит результаты дизъюнктивных членов. Для буферизованных выходов уровень 4 отображает выход Q регистра. Следующий пример использует устройство p22v10:

```
pin 1 = CLK;
pin 2 = IN2;
pin 3 = IN3;
....
pin 14 = OUT14;
pin 15 = OUT15;
....
OUT14.D = IN2;
```

```
OUT14.AR = IN3;  
OUT14.OE = IN4;
```

Файл результатов моделирования:

```
order CLK, IN2, IN3, IN4, OUT14, OUT15;  
***** before output buffer *****  
???? ..LL...0001:  
0011 ..HH.....  
*****before output buffer*****  
????  HH...0004  
C100 ...ZZ.....
```

\$EXIT

Директива **\$EXIT** используется для принудительного прерывания процесса моделирования. Тестовые векторы, следующие за данной директивой, игнорируются. Эта директива полезна при отладке буферизованных схем, в которых ложный перепад в одном векторе приводит к ошибке во всех последующих векторах.

Размещение команды **\$EXIT** после ошибочного вектора заостряет внимание на истинных проблемах вместо многочисленных ложных ошибок, возникающих вследствие неправильного перепада.

\$\$SIMOFF

Директива программы моделирования **\$\$SIMOFF** используется для выключения вычисления тестовых векторов. Тестовые векторы, стоящие после данной директивы, вычисляются только для неправильных значений и правильного номера тестовых значений. Эта директива полезна при тестировании асинхронных схем, для которых программа моделирования не может правильно вычислить значения буферизованных выходов.

\$\$SIMON

Директива **\$\$SIMON** используется для отмены действия директивы **\$\$SIMOFF**. Тестовые векторы, следующие за данной директивой, вычисляются в полном объеме.

Расширенный синтаксис

Далее следует описание команд, которые можно располагать в секции тестовых векторов **.SI** файла после ключевого слова **VECTORS**.

Оператор присваивания (\$SET)

Синтаксис: **\$SET <variable> = <constant>;**

<variable> = <single_sym> | <field> | <defined_variable>

<constant> = <quoted_val> | <tv_string>

<quoted_val> = числа, заключенные в одиночные или двойные кавычки, представляющие входы или выходы. Они будут интерпретированы с учетом установленного основания системы счисления и не должны содержать неопределенных значений.

<tv_string> = строка со значениями тестового вектора. Количество значений должно быть эквивалентно количеству бит в переменной, которой они присвоены.

Действие:

С помощью этого оператора производится присваивание константного значения символу, массиву (полю) или переменной. Он вступает в действие немедленно, но затрагивает только пользовательские значения переменной; результаты последнего шага моделирования не изменяются. Может располагаться в любом месте секции тестовых векторов.

Пример:

```
$set input = '3F'; /* single quotes for inputs */
$set output = "80"; /* double quotes for outputs */
$set Z = HHHH; /* test vector values for a 4-bit
                output variable */
```

Арифметические и логические операции (\$COMP)

Синтаксис: \$COMP <variable> = <expression>;

<variable> = <single_sym> | <field> | <defined_variable>

<expression> = любое арифметическое или логическое выражение, в котором операнды могут быть переменными или константами.

Константы должны быть десятичными числами (без кавычек), использование скобок разрешено.

Таблица 5.32

Оператор	Функция	Приоритет
!	NOT	1
&	AND	2
#	OR	3
\$	XOR	4

Таблица 5.33

Оператор	Функция	Приоритет
*	Умножение	1
/	Деление	1
+	Сложение	2
-	Вычитание	2

И логические (табл. 5.32), и арифметические операции (табл. 5.33) можно использовать в одном и том же выражении без ограничений. Логические операции имеют более высокий приоритет, однако, приоритет выполнения операций в выражении можно изменить, посредством использования скобок.

Действие:

Выполняется вычисление выражения и присваивание результата указанной переменной. При вычислении выражения используются текущие значения операндов (пользовательские значения). Вступает в действие немедленно, но затрагивает только пользовательские значения переменной; результаты последнего шага моделирования не изменяются. Может располагаться в любом месте секции тестовых векторов.

Примеры:

```
$COMP A = (!B + C) * A + 1;
$COMP X = (Z / 2) # MASK;
```

Генерация тестового вектора (\$OUT)

Синтаксис: \$OUT

Действие:

Запускает процесс моделирования для текущих значений символов и генерирует тестовый вектор. Полезно использовать после команд **\$set** и **\$comp**, что позволяет задействовать в вычислении вектора присвоенные в предыдущих операторах значения.

Пример:

Показанный ниже набор команд в SI файле:

```
ORDER: _CLOCK, %3, _OE, %3, shift, %1, input, %2, output;
VECTORS:
0 0 'X' XXXXXXXX LLLLLLLL /* power-on reset state */
$set _CLOCK = C;
$set shift = '0';
$set input = '80';
$set output = "80";
$out;
```

произведет следующий результат в SO файле:

```
0001: 0 0 XXX XXXXXXXX LLLLLLLL
0002: C 0 000 10000000 HLLLLLLL
```

Условный оператор (\$IF)

```
Синтаксис:  $IF <condition> :  
            <block_1>  
            [ $ELSE :  
              <block_2> ]  
            $ENDIF;
```

<condition> = <var_list> <logic_operators> <constant>

логические операции:

=	равно
#	не равно
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно

<constant> = <quoted_val> | <tv_string>

<block_1>, <block_2> = любая последовательность операторов, включая тестовые векторы.

Использования оператора **\$ELSE** не является обязательным.

Действие:

Вычисление условия производится с использованием текущего значения переменной. Если результат равен значению **ИСТИНА**, то выполняются операторы из **<block_1>**, иначе, если присутствует ветвление **\$ELSE**, выполняются операторы из **<block_2>**. **\$ENDIF** отмечает конец оператора **IF**.

Операторы циклов

Оператор FOR

```
Синтаксис:  $FOR <count> = <n1>..<n2> :  
            <block>  
            $ENDF;
```

<count> = счетчик цикла FOR; он принимает значения между <n1> и <n2>.

<n1>, <n2> = граничные значения счетчика <count>; должны быть положительными десятичными числами.

<block> = любая последовательность операторов, включая тестовые векторы.

Действие:

Шаг 1. Инициализация счетчика <count> значением <n1>.

Шаг 2. Выполнение <block>.

Шаг 3. Если <count>=<n2> STOP.

иначе <count> увеличивается на 1 (если <n1> меньше <n2>) или уменьшается на 1 (если <n1> больше <n2>), затем повторяются шаги 2 и 3.

Оператор WHILE

```
Синтаксис:   $WHILE <condition> :  
             <block>  
             $ENDW;
```

<condition> = то же, что и в операторе IF.

<block> = любая последовательность операторов.

Действие:

Шаг 1. Вычисление условия <condition>, если ЛОЖЬ, то STOP.

Шаг 2. Выполнение <block>.

Шаг 3. Продолжить с шага 1.

Оператор DO .. UNTIL

```
Синтаксис:   $DO:  
             <block>  
             $UNTIL <condition> ;
```

<condition> = то же, что и в операторе IF.

<block> = любая последовательность операторов.

Действие:

Шаг 1. Выполнение <block>.

Шаг 2. Вычисление условия <condition>, если ИСТИНА, то STOP, иначе продолжить с шага 1.

◆ Условный оператор IF и операторы цикла могут быть вложенными, однако, максимальное количество вложений равно 10.

Операторы MACRO и CALL

Определение макроса

```
Синтаксис:   $MACRO name (<arg_list>);  
             <macro_body>  
             $MEND;
```

name = имя макроса.

<arg_list> = symbolic names, separated by commas

<macro_body> = любая последовательность операторов, за исключение вызова макроса MACRO.

Имена параметров могут находиться в любом месте тела макроса, где разрешены имена переменных или константы. Они не могут заменять операторы, специальные символы или зарезервированные слова.

Вызов макроса

Синтаксис: `$CALL name(<act_arg_list>);`

name = имя ранее определенного макроса.

<act_arg_list> = список фактических параметров.

Фактическими параметрами могут являться имена переменных, константы или даже параметры макроса, если оператор **CALL** расположен в теле другого макроса.

Действие:

При вызове макроса выполняются операторы, формирующие его тело, с заменой формальных параметров макроса на соответствующие фактические.

Для успешного завершения оператора вызова макроса необходимо проверить, соответствуют ли фактические параметры синтаксису тела макроса, то есть не возникнет ли синтаксическая ошибка вследствие подстановки фактических параметров.

Пример:

```
$MACRO m1(a,b,c);           /* Macro definition */
$set shift = a;
$set shift = b;
$set output = c;
$MEND;

$CALL m1('0','80',*****); /* Macro call */
```

Будут выполнены следующие операторы:

```
$set shift = '0';
$set shift = '80';
$set output = *****;
```

Далее приведен полный пример того, как работают эти операторы и как они могут помочь пользователю провести моделирование своего проекта без ввода большого количества тестовых векторов.

Эти два SI файла произведут следующий результат:

1. Старый способ:

Name	Barrel22;
Partno	CA0006;
Date	05/11/89;
Revision	02;
Designer	Kahl;
Company	Protel International;
Assembly	None;

```

Location      None;
Device        g20v8a;

ORDER:  _CLOCK, %3, _OE, %3, shift, %1, input, %2, output;
VECTORS:
0  0 'X' XXXXXXXX  HHHHHHHH  /* power-on reset state */
C  0 '0' 10000000  HLLLLLLL  /* shift 0 */
C  0 '1' 10000000  LHLLLLLL  /* shift 1 */
C  0 '2' 10000000  LLHLLLLL  /* shift 2 */
C  0 '3' 10000000  LLLHLLLL  /* shift 3 */
C  0 '4' 10000000  LLLLHLLL  /* shift 4 */
C  0 '5' 10000000  LLLLHLHL  /* shift 5 */
C  0 '6' 10000000  LLLLLLHL  /* shift 6 */
C  0 '7' 10000000  LLLLLLLH  /* shift 7 */
C  0 '0' 01111111  LHHHHHHH  /* shift 0 */
C  0 '1' 01111111  HLHHHHHH  /* shift 1 */
C  0 '2' 01111111  HHLHHHHH  /* shift 2 */
C  0 '3' 01111111  HHHLHHHH  /* shift 3 */
C  0 '4' 01111111  HHHHLHHH  /* shift 4 */
C  0 '5' 01111111  HHHHHLHH  /* shift 5 */
C  0 '6' 01111111  HHHHHHLH  /* shift 6 */
C  0 '7' 01111111  HHHHHHHL  /* shift 7 */

```

2. Новый способ:

```

ORDER:  _CLOCK, %3, _OE, %3, shift, %1, input, %2, output;
VECTORS:
0  0 'X' XXXXXXXX  LLLLLLLL /* power-on reset state */
$set _CLOCK = C;
$set shift = '0';
$set input = '80';
$set output = "80";
$for i = 1..16 :
$out;
$if shift = '7':
$set shift = '0';
$set input = '7f';
$set output = "7f";
$else:
$comp shift = shift + 1;
$comp output = output / 2;
$if input = '7f':
$comp output = output # 128;
$endif;
$endif;
$endF;

```

или с использованием макроса:

```
ORDER: _CLOCK, %3, _OE, %3, shift, %1, input, %2, output;
VECTORS:
$macro m1(x,y,z);
$set shift = x;
$set input = y;
$set output = z;
$mend;

$macro m2(a,b,c,d);
$call m1(a,b,c);
$for i = 1..8 :
$out; $comp shift = shift + 1;
$comp output = output / 2 + d;
$endf;
$mend;
0 0 'X' XXXXXXXX LLLLLLLL /* power-on reset state */
$set _CLOCK = C;
$call m2('0','80',"80", 0);
$call m2('0','7f',"7f", 128);
```

3. Выходной файл:

```
CSIM: CUPL Simulation Program Version
4.2a Serial# ...
Copyright (c) 1996 Protel International
CREATED Wed Dec 04 03:00:11 1991
LISTING FOR SIMULATION FILE: barrel22.si
1: Name           Barrel22;
2: Partno         CA0006;
3: Date           05/11/89;
4: Revision       02;
5: Designer       Kahl;
6: Company        Protel International;
7: Assembly       None;
8: Location       None;
9: Device         g20v8a;
10:
11: FIELD input  = [D7,D6,D5,D4,D3,D2,D1,D0];
12: FIELD output = [Q7,Q6,Q5,Q4,Q3,Q2,Q1,Q0];
13: FIELD shift  = [S2,S1,S0];
14:
15: ORDER: _CLOCK, %3, _OE, %3, shift, %1, input, %2, output;
```

```

16:
17: var X = Q7;
18: var Y = Q7..4;
19:
=====
      _
      C
      L
      O
      C  _
      C  O  shi
      K  E  ft  input      output
=====
0001: 0  0  XXX XXXXXXXX  LLLLLLLL
0002: C  0  000 10000000  HLLLLLLL
0003: C  0  001 10000000  LHLLLLLL
0004: C  0  010 10000000  LLHLLLLL
0005: C  0  011 10000000  LLLHLLLL
0006: C  0  100 10000000  LLLLHLLL
0007: C  0  101 10000000  LLLLHLLL
0008: C  0  110 10000000  LLLLHLHL
0009: C  0  111 10000000  LLLLHLHL
0010: C  0  000 01111111  LHHHHHHH
0011: C  0  001 01111111  HLHHHHHH
0012: C  0  010 01111111  HHLLHHHH
0013: C  0  011 01111111  HHHLLHHH
0014: C  0  100 01111111  HHHHLHHH
0015: C  0  101 01111111  HHHHLLHH
0016: C  0  110 01111111  HHHHHLLH
0017: C  0  111 01111111  HHHHHHLL

```

Имеется одно обстоятельство, которое надо всегда учитывать при создании входного файла моделирования новым синтаксическим способом:

Если перед операторами проверки условий (**IF**, **WHILE**, **UNTIL**) стоят один или несколько операторов **\$SET** или **\$COMP** без промежуточного оператора **\$OUT**, то задаваемые ими значения не будут учитываться при проверке условий, так как условия проверяются только для значений используемых переменных, полученных в результате последнего моделирования.

Например, вы хотите получить на выходе следующий результат:

```

ORDER: _CLOCK,clr,dir,!_OE,%2,count,%1,carry;
var mode = clr,dir;
VECTORS:
C 100          LLLL L  /* synchronous clear to state 0 */
C 000          LLH L  /* count up to state 1 */
C 000          LLHL L /* count up to state 2 */

```

```
C 000          LLHH L /* count up to state 3          */
C 000          LHLL L /* count up to state 4          */
C 000          LHLH L /* count up to state 5          */
C 000          LHHL L /* count up to state 6          */
C 000          LHHH L /* count up to state 7          */
C 000          HLLL L /* count up to state 8          */
C 000          HLLH H /* count up to state 9 - carry */
```

Приведенная ниже последовательность операторов будет генерировать неправильный результат:

```
ORDER: _CLOCK,clr,dir,!_OE,%2,count,%1,carry;
var mode = clr,dir;
VECTORS:
C 100 LLLL L $set mode = '0';
$for i=1..9 :
$comp count = count + 1;
$if count="9":
$set carry = H;
$endif;
$out;
$endf;
```

который будет иметь вид:

```
0001: C 100 LLLL L
0002: C 000 LLLH L
0003: C 000 LLHL L
0004: C 000 LLHH L
0005: C 000 LHLL L
0006: C 000 LHLH L
0007: C 000 LHHH L
0008: C 000 LHHH L
0009: C 000 HLLL L
0010: C 000 HLLH H
      ^
[0019sa] user expected (L) for carry
```

Неправильный результат был получен из-за того, что значение счетчика **count**, используемое при проверке условия в операторе **IF** для вектора 10, в то время было текущим смоделированным значением (показанным в векторе 9) и не устанавливалось с помощью команды **\$comp**.

Правильная последовательность операторов будет выглядеть так:

```
C 100 LLLL L
$set mode = '0';
$for i=1..9 :
$if count="8":
$set carry = H;
$endif;
$comp count = count + 1;
$out;
$endif;
```

Виртуальное моделирование

Виртуальное моделирование позволяет создавать и моделировать проект без указания целевой микросхемы. Таким образом, появляется возможность получить работающий проект еще до выбора архитектуры, на которой он будет осуществлен. Это особенно полезно для разработок, которые в последствии могут быть разделены на несколько отдельных частей.

Применение виртуального моделирования не вызывает особых сложностей. Нет необходимости в изучении новых команд и синтаксиса, просто нужно использовать мнемонические обозначения виртуальных (**VIRTUAL**) устройств при компиляции и моделировании для выполнения программы виртуального моделирования.

Программа виртуального моделирования также используется для моделирования проектов на базе устройств FPGA. Если полное архитектурное моделирование невозможно по причинам прав собственности внутренних блоков схемы или из-за сложности внутренних логических ресурсов, тогда виртуальное моделирование является лучшим вариантом на стадии тестирования схемы.

Моделирование неисправностей

Для любых конъюнктивных членов может быть смоделирована внутренняя ошибка, что позволяет определить зону действия такой ошибки. Формат оператора:

```
STUCKL n ;
```

или

```
STUCKH n ;
```

где n — это номер JEDEC перемычки для первой перемычки в конъюнктивном члене.

Файл документации с расширением **.DOC** входит карта перемычек, в которой перечислены номера первых перемычек для всех конъюнктивных членов микросхемы.

Формат 1 принудительно устанавливает конъюнктивный член в 0.

Формат 2 принудительно устанавливает конъюнктивный член в 1. Команда **STUCK** должна быть расположена между операторами **ORDER** и **VECTORS**.

Просмотр смоделированных временных диаграмм сигналов

Результаты программы моделирования обычно отображаются как временные диаграммы сигналов в редакторе временных диаграмм (Waveform Editor). Файл листинга в формате ASCII с расширением **.SO** содержит результаты моделирования.

Масштаб по оси времени (Time Base)

Ось времени отображается в качестве шкалы в верхней части окна просмотра сигналов. По умолчанию изменение событий в файле списка происходит с интервалом в 1 нс.

Вертикальный маркер на шкале времени показывает текущую позицию курсора. Точное значение времени располагается в левой части строки статуса.

Название сигнала (Signal Name)

Названия сигналов извлекаются из файла списка. Названия могут быть отредактированы и для них могут быть изменены шрифты. Для редактирования названия необходимо осуществить двойной щелчок мыши по этому названию.

Значение сигнала (Signal Value)

Содержимое столбца Value отражает состояние каждого из сигналов в текущем временном интервале. Текущий временной интервал — это левая граница шкалы времени. Можно заметить, что при прокрутке временных диаграмм содержимое столбца

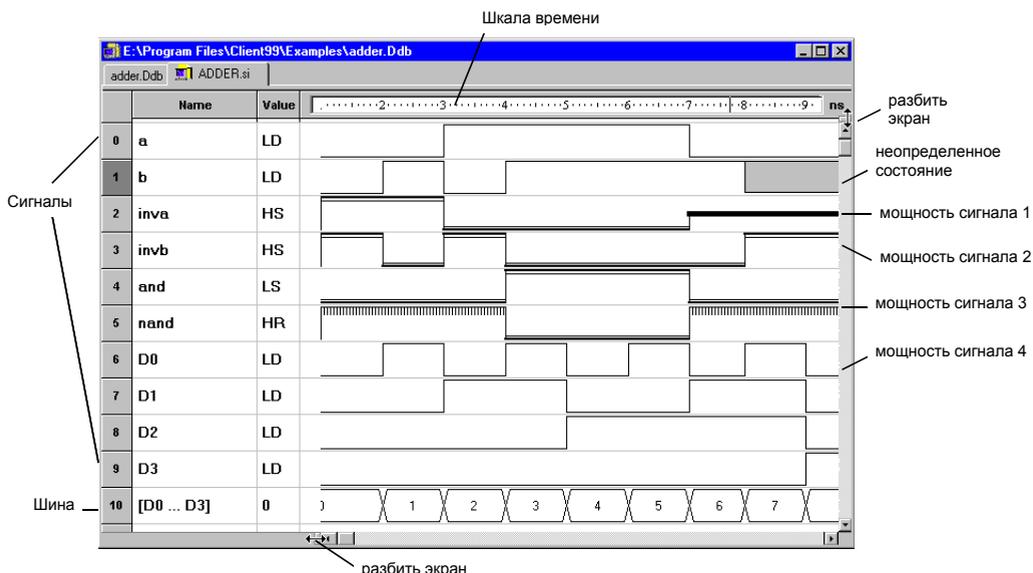


Рис. 5.17. Окно просмотра временных диаграмм сигналов.

Value изменяется.

Каждое значение состоит из двух символов, которые имеют следующий смысл:

- L Состояние сигнала — низкий уровень.
- H Состояние сигнала — высокий уровень.
- D Мощность сигнала — от проходящего сигнала.
- S Мощность сигнала — от питания.
- R Мощность сигнала — резистивная.
- xx Сигналы шины отображают текущее значение шины (в двоичном, десятичном или шестнадцатеричном виде).

Почему сигналы выглядят по-разному?

Сигналы закрашиваются по-разному в зависимости от их мощности. На предыдущей странице приведен пример такой закрашки. Для просмотра всех используемых цветов и соответствия их мощности сигналов применяется команда меню View » Legend.

Мощность отображаемого сигнала определяется из типа сигнала, приведенного в файле списка.

Временные отметки

Редактор временных диаграмм включает 10 временных отметок, которые могут быть установлены в любой точке. Для этого используется команда меню **Edit » Set Timing Marks**. Команда меню **Edit » Jump** предназначена для перехода к желаемой временной отметки (горячая клавиша: J и номер временной метки).

Создание шины

Из любого количества соседних сигналов можно создать шину. Для этого необходимо:

- С помощью мыши выбрать сигналы, которые должны быть сгруппированы в шину (при этом они подсветятся черным цветом).
- Выбрать команду меню **Insert » Bus**.

Шина будет вставлена сразу за последним выделенным сигналом. В столбце **Value** отобразится значение шины в текущем временном интервале. Каждый сигнал, включенный в шину, представляет один бит в этой шине, при этом верхний выделенный сигнал является младшим битом.

Редактирование сигнала

- Для редактирования перепада необходимо осуществить двойной щелчок мыши в месте этого перепада.
- С помощью щелчка мышью и последующего перетаскивания перепада сигнала производится графическое изменение начального времени (**Start Time**).

Изменение обзора

Редактор временных диаграмм включает несколько функций, полезных при исследовании сигналов.

Горизонтальная и вертикальная прокрутка

Для прокрутки окна сигналов в любом направлении используются полосы прокрутки. Иначе эти действия можно выполнить с помощью клавиш направления и клавиш **PageUp** и **PageDown**.

Приближение и удаление

Посредством команд меню **View » Zoom In** (горячая клавиша: +) и **View » Zoom Out** (горячая клавиша —) можно изменить масштаб временной шкалы. Эти действия связаны с текущей позицией указателя мыши, поэтому перед нажатием горячих клавиш рекомендуется расположить его в интересующем месте.

Панорамирование

Для панорамирования сигналов используется клавиша **Home**. Эта горячая клавиша также связана с положением указателя мыши, поэтому перед ее нажатием рекомендуется расположить его в нужном месте.

Создание нескольких окон просмотра

Редактор временных диаграмм имеет возможность открытия нескольких окон просмотра, позволяющий одновременно просматривать различные части сигналов. О окна просмотра относительно друг друга могут быть расположены вертикально (для одновременного просмотра на разных временных отрезках) или горизонтально (для сведения разных сигналов, которые нельзя посмотреть одновременно). На предыдущем рисунке показаны места расположения курсора для осуществления функции разделения окна. Достаточно направить указатель мыши в одно из указанных мест, нажать клавишу мыши и затем перетащить появившуюся границу окна.

Панель управления редактора временных диаграмм

С помощью команды меню можно вывести или скрыть панель управления редактора временных диаграмм. Эта панель используется для перехода от одного перепада к другому в активном сигнале, а также для скрытия и вывода сигналов.

Активный сигнал

В верхней части панели управления располагается окошко с именем активного сигнала. Кнопки Jump Transition позволяют перейти (в активном сигнале) к:

- Первому перепаду (горячие клавиши: J, F).
- Предыдущему перепаду (J, P).
- Следующему перепаду (J, N).
- Последнему перепаду (J, L).

Выбор отображаемых сигналов

Редактор временных диаграмм позволяет просматривать только лишь необходимый набор сигналов посредством скрытия ненужных. Для этого необходимо выбрать неинтересные сигналы в списке и нажать кнопку Hide на панели управления.

Для выбора отображаемых сигналов используется кнопка **Select**, расположенная на главной панели инструментов. Для удаления сигналов используется кнопка **Hide**. Выделение сигналов на панели можно производить с использованием клавиш **SHIFT** и **CTRL**.



Рис. 5.18. Выбор сигналов на отображение.

Пример сессии разработки проекта на базе ПЛИС

В этом разделе приводятся пошаговые инструкции для разработки ПЛИС устройств. Здесь показано, как выполнить проектирование одного устройства двумя разными способами: посредством разработки принципиальной схемы и CUPL кода. Также приведены рекомендации по созданию исходного файла для процесса моделирования и сам процесс моделирования ПЛИС устройств.

Этапы разработки

Проверка технического задания

Для начала рекомендуется тщательно изучить поставленную задачу. Следует помнить, что устройство может быть спроектировано либо посредством создания принципиальной схемы, либо посредством описания на языке CUPL. При выборе описания на языке CUPL главными инструментами проектирования будут конечные автоматы, Булевы уравнения и таблицы истинности. Надо попытаться определить, какой тип синтаксиса наилучшим образом подходит для выполнения данной задачи.

Создание исходного файла для компилятора

Для создания исходных файлов рекомендуется использовать входящие в комплект поставки файлы шаблонов, где следует удалить ненужные секции. Далее требуется отредактировать заголовок файла таким образом, чтобы он отражал содержимое нового проекта.

Составление выражений

Для правильного описания требуемой логической схемы необходимо составить выражения, пользуясь синтаксическими правилами языка CUPL. Выражения могут быть записаны в Булевом формате, в виде конечных автоматов или как таблица истинности.

Выбор целевого устройства

При выборе целевой микросхемы следует убедиться, что выбранное устройство имеет достаточное число входных выводов. Проверьте также наличие достаточного количества буферизованных и небуферизованных выходных выводов. При необходимости, убедитесь в наличии выводов управления с тремя состояниями. Следует также проверить, может ли устройство адекватно управлять требуемым количеством комбинационных логических блоков. Выбор целевой микросхемы осуществляется в диалоговом окне **Configure PLD**.

Распределение выводов

Далее следует корректно распределить входы и выходы проекта между выводами целевого устройства. Делать это надо исходя не только из оптимальной реализации логической схемы, но и с учетом того, что конечное устройство будет располагаться

на печатной плате и соединяться с другими компонентами. Используя предоставляемые производителем справочные материалы, необходимо проверить правильность функционирования выбранного устройства.

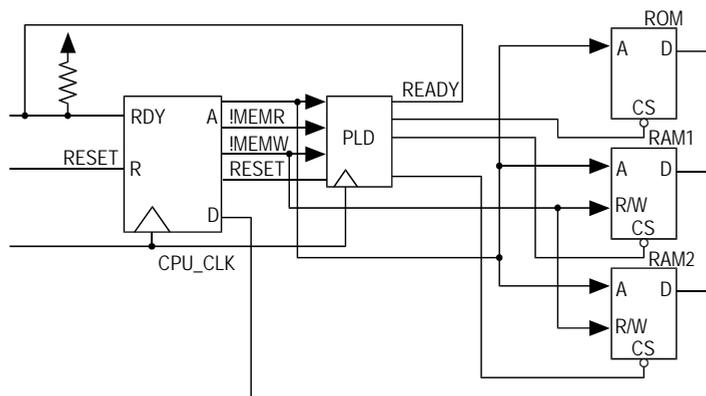


Рис. 5.19. Микропроцессорная система на базе ПЛИС

Подготовка к компилированию

Определите, какие форматы файлов будут необходимы для загрузки схемы и моделирования, а также какой минимизатор вам понадобится. Теперь все готово для компиляции проекта.

Шаг 1 — Проверка технического задания

В этом примере ПЛИС обеспечивает гибкий интерфейс между процессором и периферийными устройствами, выполняя функции временного управления и декодирования адреса для получения сигнала выбора микросхемы CS.

Карта памяти, приведенная на рисунке 5.20, показывает, где в адресном пространстве процессора находятся микросхемы ПЗУ и ОЗУ. ПЛИС будет декодировать эти диапазоны адресов для генерирования сигнала выбора соответствующей микросхемы.

Приведенная ниже таблица создается на основе карты памяти, изображенной на рисунке слева,

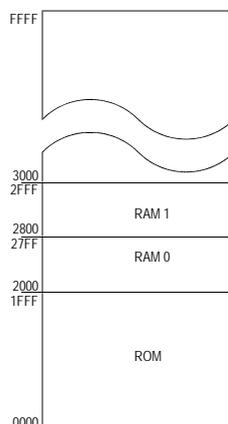


Рис. 5.20. Карта распределения памяти

где адреса показаны в шестнадцатеричной форме. В ней показано состояние адресных линий для каждого диапазона декодируемых адресов. Из таблицы следует, что ПЛИС должна в качестве входов использовать адресные линии A11 — A15.

Таблица 5.34

Микросхемы	A15	A14	A13	A12	A11	A10	A9	A8	A7-A4	A3-A0
ROM (0000-1FFF)	0	0	0	X	X	X	X	X	X	X
RAM0 (2000-27FF)	0	0	1	0	0	X	X	X	X	X
RAM1 (2800-2FFF)	0	0	1	0	1	X	X	X	X	X

Вследствие малого быстродействия микросхем ПЗУ, ПЛИС должна также уметь генерировать циклы ожидания при обращении процессора к этим микросхемам.

Взаимосвязь различных сигналов на временной диаграмме на рисунке 5.21 показана стрелками.

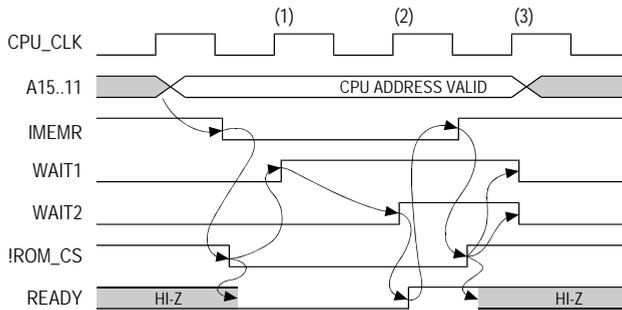


Рис. 5.21. Временные диаграммы сигналов Wait

Далее следует описание приведенной временной диаграммы. Числа в скобках показывают номер положительного перепада тактового сигнала (**CLOCK**).

Последовательность циклов ожидания начинается после установки правильного адреса и продолжается до строба чтения памяти. Необходимо рассмотреть только сигнал **!MEMR**, так как циклы ожидания генерируются только для ПЗУ.

Когда строб **!MEMR** активен для адреса, соответствующего области ПЗУ, генерируется сигнал **!ROM_CS** и буфер с тремя состояниями переключается в логический "0", что приводит к подаче низкого уровня на вывод **READY** процессора (это означает перевод процессора в состояние ожидания). Следующий положительный перепад тактового сигнала (1), идущий уже после **!ROM_CS**, устанавливает сигнал **WAIT1**. Далее после прохождения одного целого тактового периода по положительному перепаду тактового сигнала (2) устанавливается сигнал **WAIT2**, а период цикла ожидания (один такт

процессора) завершается, что приводит к установке сигнала **READY** в положение логической "1". Это заставляет процессор продолжить цикл чтения путем сброса stroba **!MEMR**. Далее сбрасывается сигнал **!ROM_CS**, который устанавливает буфер в высокоомное Z-состояние, что приводит к сбросу сигналов **WAIT1** и **WAIT2** по следующему положительному перепаду тактового сигнала (3). Теперь генератор циклов ожидания готов для следующего обращения процессора.

Шаг 2 — Создание исходного кода на языке CUPL

На этом шаге создается файл логического описания проекта на базе ПЛИС. Этот файл пишется на языке описания логических схем CUPL. Такой файл является входным для компилятора, который, обработав его, создает файл для загрузки в программатор.

Создание заголовочной секции исходного файла проекта на базе ПЛИС приведено ниже. Следующая за заголовком секция — титульный блок, который в действительности содержит обычные комментарии, обрамленные наборами символов **/*** и ***/**. В титульном блоке приводится краткая информация, описывающая данный проект.

```

                                WAITGEN.PLD
Name                            WaitGen;
Partno                          P9000183;
Date                             07/16/87;
Revision                         02;
Designer                         Osann;
Company                          ATI;
Assembly                         PC Memory;
Location                         U106;
Device                           ; /* the device is selected later */
/*****
/* This device generates chip select signals for one */
/* 8Kx8 ROM and two 2Kx8 static RAMs. It also drives */
/* the system READY line to insert a wait-state of at */
/* least one cpu clock for ROM accesses             */
*****/

```

Шаг 3 — Составление выражений

Для упрощения ввода специфических выражений для декодирования адреса и генерации циклов ожидания рекомендуется сначала составить выражения для промежуточных переменных. Имена таким переменным назначаются произвольно, то есть им не будут соответствовать какие-либо выводы. Промежуточные выражения вводятся в файле **WAITGEN.PLD**, предназначенном специально для этих целей.

Первое вводимое промежуточное выражение представляет собой объявление битового поля, которое определяет шину адреса. Используем для этого имя **MEMADR** (от английского **memory address** — адрес памяти) для представления адреса и введем следующее выражение:

```
FIELD MEMADR = [A15..11] ;
```

На системной диаграмме, приведенной выше, можно заметить, что установка сигналов выбора микросхемы для статического ОЗУ зависит не только от адреса, но и от стробов данных **MEMW** и **MEMR**.

Для упрощения выражений для сигналов выбора микросхемы при обращении к статическому ОЗУ создадим сигнал **MEMREQ** (от английского **memory request** — запрос памяти). Введем следующее выражение:

```
MEMREQ = MEMW # MEMR ;
```

Всякий раз, когда в других выражениях будет встречаться **MEMREQ**, компилятор будет осуществлять замену этой переменной на выражение **MEMW # MEMR**.

Отметим, что на временных диаграммах декодирование адреса, соответствующего ПЗУ, сочетается со стробом **!MEMR** для активизации сигнала выбора микросхемы ПЗУ (**ROM_CS**) и инициации последовательности циклов ожидания.

Создадим промежуточную переменную **SELECT_ROM**, представляющую комбинацию строба **!MEMR** и адреса, попадающего в диапазон, отведенный для ПЗУ. Для этого введем следующее:

```
SELECT_ROM = MEMR & MEMADR : [0000..1FFF] ;
```

После определения этой промежуточной переменной можно задать выражения для декодирования адреса и генерации циклов ожидания.

Если сигнал **ROM_CS**, который возвращается в матрицу, используется для инициации циклов ожидания, то в ПЛИС появляется дополнительная задержка распространения. Так как в нашем примере тактовая частота относительно невелика (4 — 8 МГц), эта дополнительная задержка не вызовет проблем. Однако, при использовании высоких тактовых частот рекомендуется создать еще один подобный логический блок (используя промежуточную переменную **SELECT_ROM**), применив выражения для буферизированной логической схемы.

Создадим сигнал выбора микросхемы ПЗУ **ROM_CS**, используя промежуточную переменную **SELECT_ROM**:

```
ROM_CS = SELECT_ROM ;
```

Сигналы выбора микросхем ОЗУ (**RAM_CS0** и **RAM_CS1**) зависят от состояния **MEMREQ** и адресной шины. Для декодирования этих сигналов используем операцию **CUPL** для выбора желаемого диапазона адресов с помощью указания верхней и нижней границы диапазона:

```
RAM_CS0 = MEMREQ & MEMADR : [2000..27FF];
RAM_CS1 = MEMREQ & MEMADR : [2800..2FFF];
```

Далее создадим выражения, которые связывают генерацию и временные соотношения циклов ожидания. Сначала, как показано на временной диаграмме, требуется, чтобы при подаче сигнала выбора микросхемы ПЗУ сигнал **WAIT1** устанавливался по следующему фронту тактовых импульсов процессора. Согласно правилам работы D-триггера логический уровень на его входе передается на его выход Q по приходу так-

тогового импульса. Введем выражение для этого сигнала, где **WAIT1.D** представляет сигнал на входе D-триггера, реализованного в устройстве ПЛИС:

```
WAIT1.D = SELECT_ROM & !RESET ;
```

Заметим, что в выражении для **WAIT1.D** сигнал **!RESET** подвергается логическому умножению (**AND**) с остальной частью выражения для того, чтобы выполнить синхронный сброс при подаче сигнала **RESET**.

Теперь создадим сигнал **WAIT2**, который генерируется по тактовому импульсу, идущему после установки сигнала **WAIT1**. Поэтому выражение для **WAIT2.D** будет зависеть от сигнала **WAIT1**. Так как **WAIT2.D** должен сбрасываться по тактовому импульсу, идущему после снятия сигнала доступа процессора к ПЗУ, необходимо подвергнуть логическому умножению (**AND**) переменные **SELECT_ROM** и **WAIT1**:

```
WAIT2.D = SELECT_ROM & WAIT1 ;
```

Создадим сигнал **SELECT_ROM** согласно временной диаграмме, чтобы показать, что буфер с тремя состояниями должен быть включен, когда декодируется адрес ПЗУ и строб данных **MEMR** активен. Таким образом, введем следующее выражение для выхода буфера с тремя состояниями:

```
READY.OE = SELECT_ROM ;
```

Данное выражение определяет, когда буфер с тремя состояниями активизирует свой выход и переходит из Z-состояния, но оно не определяет, каким логическим уровнем управляется сигнал. Выражение для **READY** определяет этот логический уровень — сигнал должен оставаться неактивным до тех пор, пока полный период цикла ожидания не будет равен одному полному тактовому циклу процессора. Это условие не будет выполнено, пока не установится сигнал **WAIT2**, поэтому выражение для **READY** выглядит следующим образом:

```
READY = WAIT2;
```

Шаг 4 — Выбор целевого устройства

Следующим шагом после составления всех выражений является определение совместимости микросхем ПЛИС, выпускаемых разными производителями. Основными пунктами рассмотрения при выборе целевого устройства будут:

- Требуемое количество входных выводов.
- Относительное количество буферизованных и небуферизованных выходных выводов.
- Наличие выводов с третьим состоянием (если необходимо).
- Требуемое количество комбинационных логических блоков (термов производства) для выполнения логических функций каждого выражения.

Самым простым устройством, удовлетворяющим требованиям данного примера, является микросхема PAL16R4, также здесь может быть использована микросхема GAL16v8. На рисунке 5.22 приведено графическое представление устройства на базе ПЛИС с распределением выводов.

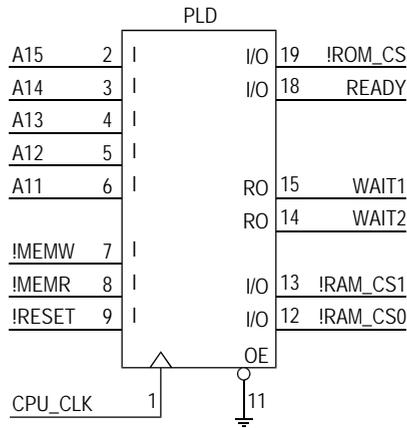


Рис. 5.22. Распределение выводов ПЛИС.

Заметим, что при распределении выводов трем сигналам выбора микросхемы CS назначены выходы типа I/O, которые всегда будут работать в качестве выходов. Вывод **READY**, который подключается к шине процессора для генерации сигнала **READY**, функционирует в режиме с тремя состояниями. Двум D-триггерам, выполняющим функцию генератора циклов ожидания, назначены выходные выходы, которые подключены к внутреннему регистру.

Один из этих буферизованных выводов можно использовать непосредственно для подачи сигнала **READY**, так как логическая функция этого сигнала такая же, как и у сигнала **WAIT2**. Однако, тогда потребовалось бы использование отдельного сигнала разрешения с тремя состояниями, подключенного к выводу 11 целевого устройства. А, так как вывод 11 управляет выходами всех четырех выводов, соединенных с внутренними регистрами, то это делает невозможным использование двух других буферизованных выходных выводов в целях, не задействованных для генерации циклов ожидания.

Рекомендуем в этой ситуации не использовать управление с помощью отдельного выхода с тремя состояниями, так как трудно предугадать все изменения, которые могут быть сделаны во время дальнейшего развития проекта. Поэтому вывод 11 соединяется с землей, благодаря чему выходы регистров с тремя состояниями подключаются к выводам устройства.

Микросхема PAL16R4 имеет по крайней мере семь доступных конъюнктивных членов для каждого выхода, и этого вполне достаточно для данной задачи. В качестве альтернативы может быть использована микросхема GAL16V8.

Когда файл **WAITGEN.PLD** является активным документом, необходимо выбрать команду меню **PLD » Configure**, затем нажать кнопку **Target Device Change** (изменить целевое устройство) для вызова диалогового окна **Target Device** (целевое устройство). Далее в списке **Device Type** (тип устройства) найти и выбрать строку **GAL**, и затем в

списке **Device Name** (имя устройства) выбрать строку g16v8. Для закрытия обоих диалоговых окон нажмите в каждом из них кнопку **ОК**. Теперь строка **DEVICE** в заголовке исходного файла ПЛИС будет выглядеть так:

```
Device      g16v8;
```

Шаг 5 — Распределение выводов

Выполним распределение выводов таким образом, чтобы оно соответствовало приведенному ранее рисунку.

Для обеспечения согласования проекта с реальным устройством при выполнении распределения выводов нужно убедиться, что полярность и уровни сигналов не противоречат сигналам на логической схеме. Далее приведено распределение выводов:

```

                                WAITGEN PIN ASSIGNMENTS
/**   Inputs   **/
Pin 1      = cpu_clk   ;          /* CPU clock                */
Pin [2..6] = [a15..11] ;          /* CPU Address Bus         */
Pin [7,8]  = ![memw,memr] ;      /* Memory Data Strobes    */
Pin 9      = reset    ;          /* System Reset           */
Pin 11     = !oe      ;          /* Output Enable          */

/**   Outputs  **/
Pin 19     = !rom_cs  ;          /* ROM Chip select        */
Pin 18     = ready   ;          /* CPU Ready signal       */
Pin 15     = wait1   ;          /* Start wait state       */
Pin 14     = wait2   ;          /* End wait state         */
Pin [13,12] = ![ram_cs1..0] ;    /* RAM Chip selects       */

```

Законченный вид файла описания:

```

WAITGEN.PLD
Name                WaitGen;
Partno              P9000183;
Date                07/16/87;
Revision            02;
Designer            Osann;
Company             ATI;
Assembly            PC Memory;
Device              g16v8;

/*****
/* This device generates chip select signals for one */
/* 8Kx8 ROM and two 2Kx8 static RAMs. It also drives */
/* the system READY line to insert a wait-state of at */

```

```

/* least one cpu clock for ROM accesses */
/*****
/*****
/** Allowable Target Device Types : GAL16V8, PAL16R4 */
/*****
/** Inputs */
Pin 1 = cpu_clk; /* CPU clock */
Pin [2..6] = [a15..11]; /* CPU Address Bus */
Pin [7,8] = ![memw,memr] ;
/* Memory Data Strobes (active low)*/
Pin 9 = reset; /* System Reset */
Pin 11 = !oe; /* Output Enable (active low) */

/** Outputs */
Pin 19 = !rom_cs; /* ROM chip select (active low)*/
Pin 18 = ready ; /* CPU ready */
Pin 15 = wait1 ; /* Wait state 1 */
Pin 14 = wait2 ; /* Wait state 2 */
Pin [13,12] = ![ram_cs1..0] ;
/* RAM chip select (active low) */

/* Declarations and Intermediate Variable Definitions */
Field memadr = [a15..11] ; /* Give the address bus */
/* the Name "memadr" */

memreq = memw # memr ; /* Create the intermediate */
/* variable "memreq" */
select_rom = memr & memadr:[0000..1FFF] ; /* = rom_cs */

/** Logic Equations */
rom_cs = select_rom;
ram_cs0 = memreq & memadr:[2000..27FF] ;
ram_cs1 = memreq & memadr:[2800..2FFF] ;

/* read as: when select_rom is true and reset is false */
wait1.d = select_rom & !reset ;

/* read as: when when select_rom is true and wait1 is true */
/* Synchronous Reset */
wait2.d = select_rom & wait1 ; /* wait1 delayed */
*/

ready.oe = select_rom ; /* Turn Buffer off */
ready = wait2 ; /* end wait */

```

Шаг 6 — Компиляция исходного файла

На этом шаге для целевого устройства GAL20V8 компилируется файл логического описания **WAITGEN.PLD**.

Определим целевую ПЛИС и другие опции компилятора в диалоговом окне **Configure PLD**. Здесь доступны следующие опции:

- **Absolute ABS** — создание файла **WAITGEN.ABS**. Это файл абсолютных величин (модулей), который используется программой моделирования (смотри Шаг 8). Он содержит сжатое представление логических функций, которые должны быть запрограммированы в устройстве. Программа моделирования сравнивает это представление с тестовыми векторами, находящимися в созданном пользователем входном файле, и определяет являются ли векторы откликов в этом входном файле корректными для векторов входных воздействий.
- **Fuse Plot in Doc File and Equations in Doc File** — создание файла **WAITGEN.DOC**. Это файл документации. Он содержит СКНФ (совершенная конъюнктивная нормальная форма) для промежуточных переменных и переменных выходных выводов, а также карту пережигания и схему кристалла.
- **Error List LST** — создание файла **WAITGEN.LST**. Этот файл повторяет файл описания с добавлением номеров строк, а также в конец файла добавляются некоторые сообщения об ошибках, которые генерируются во время процесса компилирования.
- **JEDEC** — создание файла **WAITGEN.JED**, предназначенного для загрузки в программатор и содержащего шаблон прошивки микросхемы. В процессе моделирования в этот файл будут добавлены тестовые вектора. Имя JEDEC файла определяется в поле **NAME**, находящегося в секции заголовка.

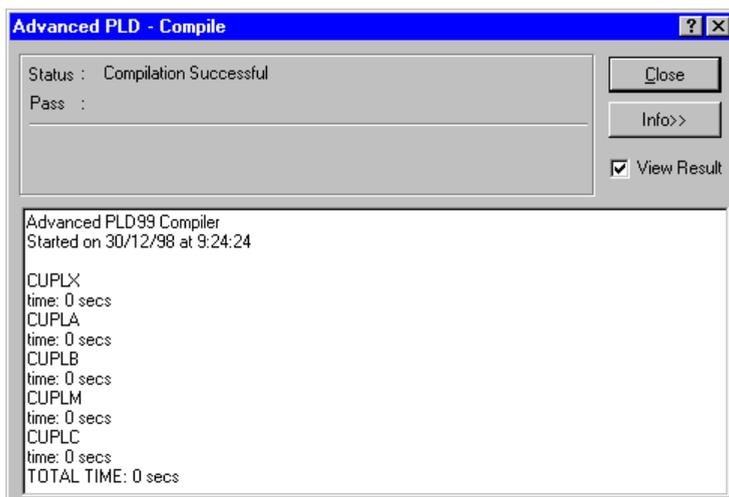


Рис. 5.23. Компиляция проекта.

После задания конфигурации компилятора необходимо нажать кнопку **OK** в диалоговом окне **Configure PLD**.

Для компиляции проекта нужно сделать документ **WAITGEN.PLD** активным и нажать кнопку **Compile** на панели инструментов **PldTools**.

Появится диалоговое окно **PLD — Compile**. При нажатии на кнопку **Info** в окне появятся сообщения, показывающие сколько времени выполняется каждый модуль компилятора. Действительное время будет зависеть от используемой конфигурации компьютера.

Компилятор создаст все заданные файлы. Если в окне был установлен флаг **View Result**, файлы **LST**, **DOC** и **JEDEC** будут автоматически открыты.

Файл **WAITGEN.LST** по существу повторяет файл-источник, добавляя в него номера строк и сообщения об ошибках. Номера строк ускоряют поиск источника той или иной ошибки, обнаруженной компилятором.

Файл документации **WAITGEN.DOC** содержит СКНФ, символьную таблицу, карту переключения и схему кристалла с расположением выводов.

Шаг 7 — Создание проекта на основе принципиальной схемы

Разработка проекта на базе ПЛИС на основе принципиальной схемы основывается на создании схемы из условных графических обозначений элементов, содержащихся в библиотеке **PLD Symbol.lib**, которая располагается в базе данных **Program Files\Design Explorer 99 SE\Library\Pld.ddb**. Эта библиотека включает обширный набор элементов PLD, CPLD и FPGA устройств. Важно, чтобы разработчик был знаком с функциональными возможностями целевых устройств до начала разработки проекта, то есть гарантировал, что проект включает только те элементы, которые могут быть реализованы в выбранном устройстве.

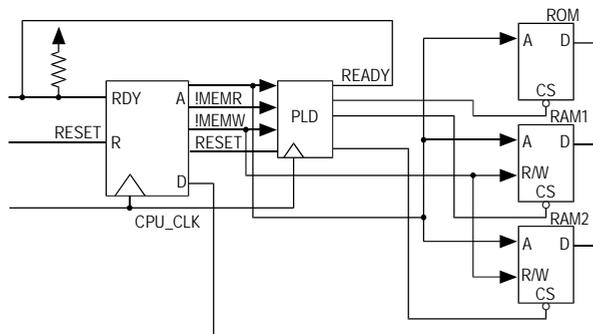


Рис. 5.23. Принципиальная схема

Логическая схема генерации сигнала выбора микросхемы

Существует большое количество способов построения логической схемы генерации сигнала выбора микросхемы. Приводимый пример использует наиболее простой для понимания способ.

Обратимся к таблице 5.34, где показано состояние адресных линий для каждого сигнала выбора микросхемы, который генерируется ПЛИС.

Используя информацию из таблицы и из схемы, приведенной на рисунке, можно написать следующие основные выражения:

```
ROM_CS = !A15 & !A14 & A13 & !MEMR
```

```
RAM_CS0 = !A15 & !A14 & A13 & !A12 & !A11 & (!MEMR # !MEMW)
```

```
RAM_CS1 = !A15 & !A14 & A13 & !A12 & A11 & (!MEMR # !MEMW)
```

Невозможно реализовать выражения для выбора микросхем ОЗУ, поскольку они содержат и логическое умножение (**AND**), и логическое сложение (**OR**), поэтому нужно сначала создать промежуточные сигналы.

Из таблицы видно, что все сигналы выбора микросхем требуют логического нуля на A15 и A14. Они могут быть объединены с помощью библиотечного элемента AND2B2:

```
ADR_15_14 = !A15 & !A14
```

(ADR_15_14 вырабатывается, когда и A15, и A14 имеют логический "0")

Сигналы выбора микросхем ОЗУ требуют логической "1" на A13 и логического "0" на A12. Поэтому эти сигналы могут быть объединены с помощью элемента AND2B1:

```
ADR_13_12 = !A13 & A12
```

(ADR_13_12 вырабатывается, когда A13 в "1", а A12 в "0")

Сигналы чтения из и записи в память также могут быть использованы для создания промежуточной переменной запроса памяти, которая будет входить в выражения для выработки сигналов выбора микросхем ОЗУ. Их можно объединить посредством элемента OR2B2:

```
MEMREQ = !MEMR & !MEMW
```

(**MEMREQ** вырабатывается, когда либо **MEMR**, либо **MEMW** устанавливается в "0")

Теперь эти промежуточные сигналы могут быть объединены с другими адресными линиями для создания трех сигналов выбора микросхемы. Получившаяся схема приведена на рисунке 5.24.

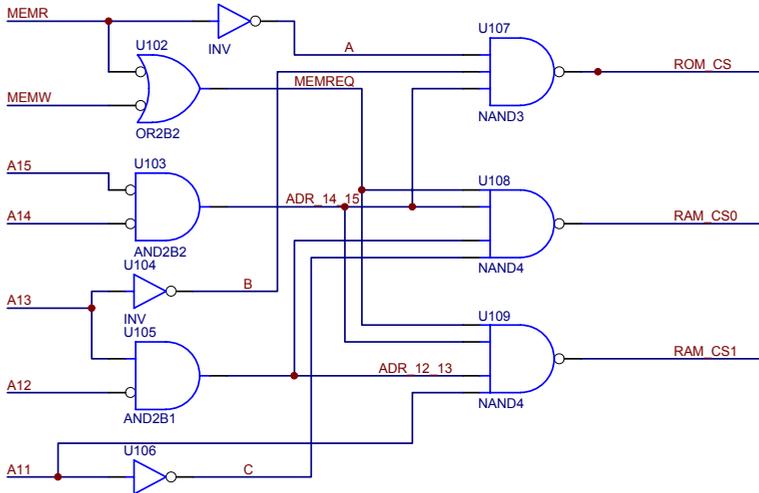


Рис. 5.24. Схема выбора микросхемы памяти.

Логическая схема для генерации циклов ожидания

Для описания логической схемы генерации циклов ожидания необходимо обратиться к временной диаграмме, представленной на рисунке 5.25.

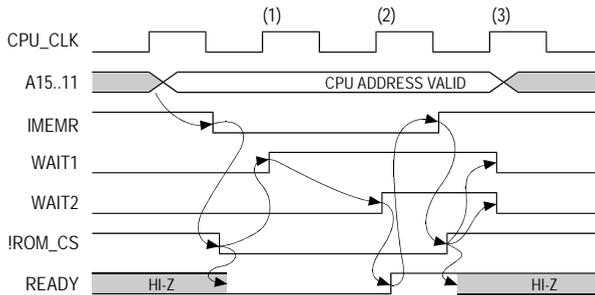


Рис. 5.25. Временные диаграммы формирования сигналов Wait

Сигнал **WAIT1**, выставляемый по ближайшему фронту тактовых импульсов, требуется в качестве ответа на подачу сигнала **ROM_CS**. Прежде чем реализовать на D-триггере схему выдачи этого сигнала, заметим, что, как показано на системной диаграмме, схема на ПЛИС должна сбрасываться по сигналу **RESET** процессора. Сброс с активным высоким уровнем может быть включен в состав генератора циклов ожидания

посредством операции логического умножения (**AND**) с **ROM_CS**. Получим следующее выражение:

```
START_WAIT = !ROM_CS & !RESET ;
```

Генератор циклов ожидания должен обеспечить пассивное положение сигнала **READY** в течении по крайней мере одного тактового интервала. **WAIT1** станет активным по первому положительному перепаду тактового сигнала процессора после выдачи **ROM_CS**, поэтому теперь нужно сгенерировать сигнал по второму тактовому импульсу. Этого можно достичь, подав сигнал с выхода первого D-триггера на D-вход второго. Второй триггер сработает по второму тактовому импульсу. Это и будет сигнал **WAIT2**.

По временной диаграмме видно, что сигнал **READY**, приходящий на процессор, после завершения операции доступа к ПЗУ должен вернуться в третье высокоомное состояние. Для этого необходимо использовать буфер с тремя состояниями, разрешением выхода которого управляет сигнал **ROM_CS**. Схема, представленная на рисунке 5.26, демонстрирует данную реализацию генератора циклов ожидания:

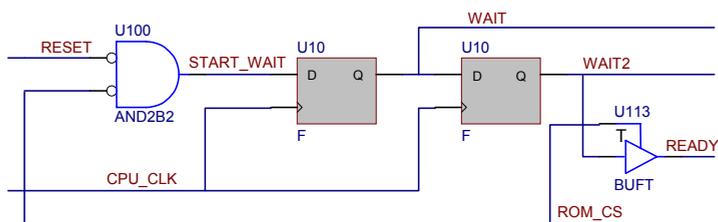


Рис. 5.26. Схема формирования сигналов ожидания.

Связь внутренней логической схемы с выводами компонента

Последним шагом в реализации проекта на базе ПЛИС является задание связей внутренней логической схемы с физическими выводами микросхемы. Для этого в библиотеке символов существуют специальные символы контактов — **IPAD**, **OPAD** и **IOPAD**. Эти символы размещаются в местах соединения схемы с физическими выводами компонента. Количество выводов, подсоединяемых к данному символу контакта, указывается в его поле **Part Field 1** и имеет следующий синтаксис:

```
LOC=PIN[pin_number]
```

После размещения соответствующих символов контактов и их нумерации согласно выбранному целевому устройству полная схема будет выглядеть, как показано на рисунке 5.27.

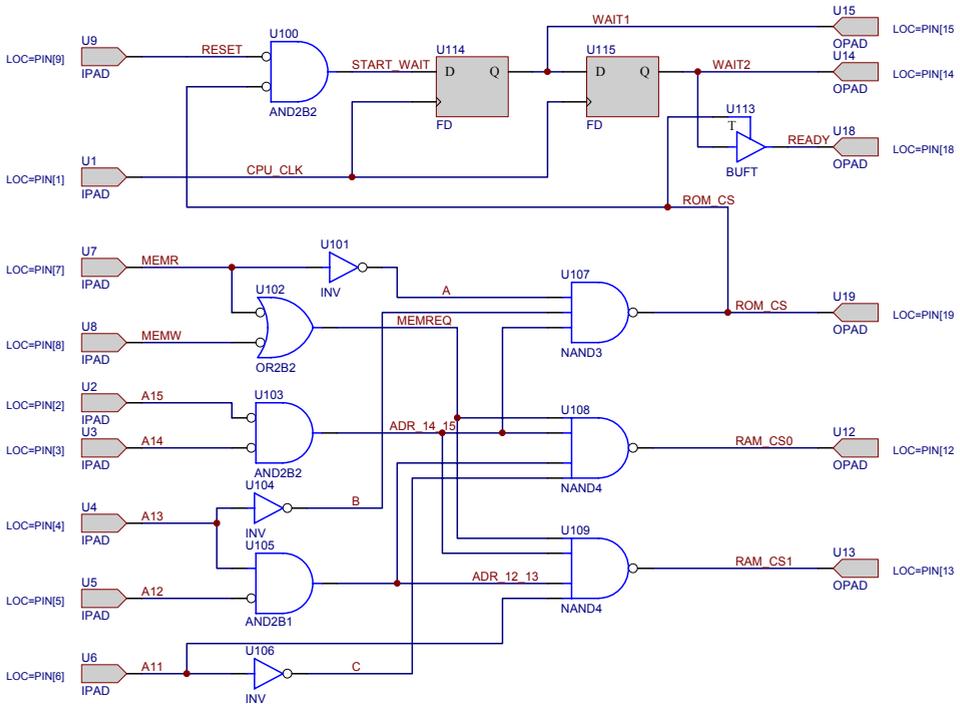


Рис. 5.27. Окончательная схема устройства.

Замечания:

- Законченный проект компилируется также, как и в случае описания его на языке CUPL (смотри Шаг 6 — компиляция исходного файла).
- Вся внутренняя логическая схема должна иметь уникальные позиционные обозначения, однако, специальных требований на них не накладывается.
- Присваивание имен всем внутренним цепям может значительно облегчить отладку устройства.
- Имена внутренних цепей должны быть уникальными и не могут быть инверсными по отношению к любым другим цепям проекта.
- Рассмотренный выше пример создания проекта на базе ПЛИС входит в стандартный комплект поставки пакета Protel 99 SE.

Шаг 8 — создание файла тестовых векторов для программы моделирования

На этом этапе выполняется моделирование функционирования скомпилированного проекта, реализованного на основе микросхемы GAL16V8, нацеленное на проверку

его правильности. Проведение данной операции непосредственно перед загрузкой данных в программатор снижает вероятность неправильного функционирования разрабатываемого устройства.

В этом разделе создается исходный файл спецификаций — **WAITGEN.SI**. Он содержит тестовые векторы, служащие входными данными для программы моделирования, которая сравнивает эти тестовые векторы и ожидаемые результаты с действительными значениями, содержащимися в файле **WAITGEN.ABS**, и сигнализирует об обнаруженных расхождениях. Файл **WAITGEN.ABS** создается во время компиляции.

Содержимое исходного файла спецификаций для проекта WaitGen.

```
Name          WaitGen;
Partno        P9000183;
Date          07/16/87;
Revision      02;
Designer      Osann;
Company       ATI;
Assembly      PC Memory;
Location      g20v8;

/*****
/* This device generates chip select signals for one */
/* 8Kx8 ROM and two 2Kx8 static RAMs. It also drives */
/* the system READY line to insert a wait-state of at */
/* least one cpu clock for ROM accesses             */
*****/

ORDER:
    cpu_clk, %2, a15, %2, a14, %2,
    a13, %2, a12, %2, a11, %2,
    !memw, %2, !memr, %2, reset, %2, !oe,
    %4, !ram_cs1, %2, !ram_cs0, %2, !rom_cs, %2,
    wait1, %2, wait2, %2, ready;

VECTORS:
/* 123456-leave six blanks to allow for numbers in .SO file */
$msg "      Power On Reset                ";
    O X X X X X 1 1 1 0   H H H * * Z
$msg "      Reset Flip Flops                ";
    C X X X X X 1 1 0 0   H H H L L Z
$msg "      Write RAM0                        ";
    0 0 0 1 0 0 0 1 0 0   H L H L L Z
$msg "      Read RAM0                         ";
    0 0 0 1 0 0 1 0 0 0   H L H L L Z
$msg "      Write RAM1                         ";
    0 0 0 1 0 1 0 1 0 0   L H H L L Z
$msg "      Read RAM1                          ";
```

```

      0 0 0 1 0 1 1 0 0 0   L H H L L Z
$msg "   Begin ROM read      ";
      0 0 0 0 0 0 1 0 0 0   H H L L L L
$msg " Two clocks for wait state, Then drive READY High  ";
$repeat2;
      C 0 0 0 0 0 1 0 0 0   H H L * * *
$msg "   End ROM Read      ";
      0 0 0 0 0 0 1 1 0 0   H H H H H Z
$msg "   End ROM Read      ";
      C 0 0 0 0 0 1 1 0 0   H H H L L Z

```

Исходный файл спецификаций состоит из трех основных частей: заголовка и титульный блок, блок оператора **ORDER** и блок оператора **VECTORS**.

Файл **WAITGEN.SI** должен иметь такой же заголовок, как и файл **WAITGEN.PLD**, что существенно обеспечит сравнение этих файлов. Поэтому рекомендуется сохранить файл **WAITGEN.PLD** как **WAITGEN.SI** и затем удалить из него все, кроме заголовка и титульного блока. Результат этой операции представлен ниже:

```

Name          WaitGen;
Partno        P9000183;
Date          07/16/87;
Revision      02;
Designer      Osann;
Company       ATI;
Assembly      PC Memory;
Location      U106;

/*****
/* This device generates chip select signals for one */
/* 8Kx8 ROM and two 2Kx8 static RAMs. It also drives */
/* the system READY line to insert a wait-state of a */
/* least one cpu clock for ROM accesses             */
*****/

```

В операторе **ORDER** перечисляются входные и выходные переменные из файла **WAITGEN.PLD**, которые должны быть включены в тестовые векторы. Порядок перечисления этих переменных должен совпадать с порядком их использования в тестовых переменных, то есть, первой должна быть переменная тактовых импульсов **CPU_CLK**, за которой следуют остальные входные переменные. Справа располагаются выходные переменные. Разделителем является символ запятой. Символ % используется для вставки пробелов между переменными. В примере, приведенном ниже, между переменными вставляется по два пробела, а между последней входной переменной **!OE** и первой выходной **!RAM_CS1** — четыре:

```
ORDER:
CPU_CLK, %2, A15, %2, A14, %2,
A13, %2, A12, %2, ALL, %2,
!MEMW, %2, !MEMR, %2, RESET, %2, !OE,
%4, !RAM_CS1, %2, !RAM_CSO, %2, !ROM_CS, %2
WAIT1, %2, READY;
```

Сразу за блоком оператора **ORDER** следует оператор **VECTORS**, который задает таблицу истинности, содержащую одиннадцать тестовых векторов. Для более простой идентификации значений в векторах для них генерируется заголовок, который размещается в **.SO** файле сразу после оператора **ORDER**. Этот заголовок состоит из символьных имен, присутствующих в операторе **ORDER**, выровненных для лучшей читаемости векторной секции.

Теперь нужно ввести тестовые векторы. Создание вектора производится с помощью присваивания некоторых значений всем входным переменным и соответствующих ожидаемых значений всем выходным переменным.

Для описания тестируемой функции устройства используется директива **\$MSG**. Оператор **ORDER**, стоящий в файле выше, определяет пространство при создании тестовых векторов. Например, создать первый вектор **Power On Reset** можно следующим образом:

```
$msg " Power On Reset      ";
0 X X X X X 1 1 1 0 H H H * * Z
```

Заметим, что выходные значения, помеченные звездочкой, используются для переменных **WAIT1** и **WAIT2**, чтобы уведомить программу моделирования о необходимости вычисления состояния регистров при включении питания, в то время, как некоторые устройства при включении питания устанавливаются в состояние **X** (неопределенное), **H** (высокое) или **L** (низкое). Использование символа звездочки дает возможность создать универсальный файл.

Далее приведены все тестовые векторы:

```
/* 123456-leave six blanks to allow for numbers in .SO file */
$msg "      Power On Reset      ";
0 X X X X X 1 1 1 0 H H H * * Z
$msg "      Reset Flip Flops    ";
C X X X X X 1 1 0 0 H H H L L Z
$msg "      Write RAM0           ";
0 0 0 1 0 0 0 1 0 0 H L H L L Z
$msg "      Read RAM0             ";
0 0 0 1 0 0 1 0 0 0 H L H L L Z
$msg "      Write RAM1           ";
0 0 0 1 0 1 0 1 0 0 L H H L L Z
$msg "      Read RAM1            ";
0 0 0 1 0 1 1 0 0 0 L H H L L Z
$msg "      Begin ROM read        ";
0 0 0 0 0 0 1 0 0 0 H H L L L L
```

```

$msg " Two clocks for wait state, Then drive READY High      ";
$repeat2;
      C 0 0 0 0 0 0 1 0 0 0      H H L * * *
$msg "      End ROM Read                                          ";
      0 0 0 0 0 0 1 1 0 0      H H H H H Z
$msg "      End ROM Read                                          ";
      C 0 0 0 0 0 0 1 1 0 0      H H H L L Z

```

Директива **\$REPEAT** в секции векторов вызывает то, что восьмой вектор будет повторен дважды. Символы звездочки в восьмом векторе для **WAIT1**, **WAIT2** и **READY** сообщают программе моделирования о необходимости вычисления выходных значений, основанных на входных значениях, и размещении результата в выходной файл.

Тактовая переменная **CPU_CLK** принимает значение "0" или "С". Значение "0" означает, что тактирование для данного вектора отсутствует. Значение "С" приводит к тому, что программа моделирования проверяет входные значения в текущем векторе, а также состояния буферизованных выходов в предыдущем векторе, которые могут посредством внутренних цепей подаваться в какие-либо точки схемы до прихода очередного тактового импульса. Затем, после прихода тактового импульса, программа моделирования вычисляет соответствующие ожидаемые значения выходов для буферизованных и не буферизованных переменных.

После ввода операторов **VECTORS** файл спецификаций необходимо сохранить. Следующим шагом будет запуск программы моделирования.

Шаг 9 — моделирование устройства

В процессе работы программа моделирования создает файл **WAITGEN.SO**, который содержит результаты моделирования. За каждым вектором следуют сообщения о найденных ошибках, и выводится имя соответствующего сигнала.

Для запуска программы моделирования необходимо сделать активным документ **WAITGEN.PLD** и нажать кнопку **Simulate**, расположенную на панели инструментов **PldTools**.

Кнопка **Info** в диалоговом окне **PLD — Simulate** используется для получения информации о ходе моделирования. Включив флаг **View Results**, можно включить автоматическую загрузку результатов моделирования в редактор сигналов **Waveform Editor**. Так как файл **WAITGEN.SO** является обычным ASCII файлом, его можно открыть любым текстовым редактором.

Содержимое файла **WAITGEN.SO**:

```

      WAITGEN.SO
CSIM:      CUPL Simulation Program
Version 4.XX Serial # XX-XXX-XXXX
copyright (c) 1996 Protel International

```

```

CREATED Thur Aug 20 09:34:16 1990
1: Name          WaitGen;
2: Partno       P9000183;
3: Date        07/16/87;
4: Revision     02;
5: Designer     Osann;
6: Company     ATI;
7: Assembly    PC Memory;
8: Location     U106;
9:
10: /*****
11: /* This device generates chip select signals for one */
12: /* 8Kx8 ROM and two 2Kx8 static RAMs. It also drives */
13: /* the system READY line to insert a wait-state of at */
14: /* least one cpu clock for ROM accesses */
15: /*****
16:
17: ORDER:
18:     cpu_clk, %2, a15, %2, a14, %2,
19:     a13, %2, a12, %2, a11, %2,
20:     !memw, %2, !memr, %2, reset, %2, !oe,
21:     %4, !ram_cs1, %2, !ram_cs0, %2, !rom_cs, %2,
22:     wait1, %2, wait2, %2, ready;
23:

```

Simulation Results

```

=====
                                     ! !
c                                     r r !
p                                     a a r
u                                     m m o w w r
_                                     _ _ m a a e
c a a a a a e e s ! c c _ i i a
l 1 1 1 1 1 m m e o s s c t t d
k 5 4 3 2 1 w r t e 1 0 s 1 2 y
=====

```

```

Power On Reset
0001: O X X X X X 1 1 1 0 H H H X X Z
Reset Flip Flops
0002: C X X X X X 1 1 0 0 H H H L L Z
Write RAM0
0003: 0 0 0 1 0 0 0 1 0 0 H L H L L Z
Read RAM0
0004: 0 0 0 1 0 0 1 0 0 0 H L H L L Z
Write RAM1
0005: 0 0 0 1 0 1 0 1 0 0 L H H L L Z
Read RAM1
0006: 0 0 0 1 0 1 1 0 0 0 L H H L L Z

```

```

Begin ROM read
0007: 0 0 0 0 0 0 1 0 0 0  H H L L L L
      Two clocks for wait state, Then drive READY High
0008: C 0 0 0 0 0 1 0 0 0  H H L H L L
0009: C 0 0 0 0 0 1 0 0 0  H H L H H H
      End ROM Read
0010: 0 0 0 0 0 0 1 1 0 0  H H H H H Z
      End ROM Read
0011: C 0 0 0 0 0 1 1 0 0  H H H L L Z

```

Сравнивая файлы **WAITGEN.SO** и **WAITGEN.SI**, можно заметить, что в результате применения директивы **\$REPEAT** созданы векторы 8 и 9, а также то, что программа моделирования заменила символы звездочки в файле **WAITGEN.SI** для сигналов **WAIT1**, **WAIT2** и **READY** соответствующими логическими уровнями (**H** — высокий и **L** — низкий).

Теперь, когда моделирование проекта успешно завершено, тестовые векторы могут быть добавлены в файл JEDEC, который был создан в результате компилирования (Шаг 6). Для этого необходимо в диалоговом окне **Configure PLD** установить опцию JEDEC и снова запустить процесс моделирования.

После этого файл **WAITGEN.JED** будет содержать как программную, так и тестовую информацию.

```

WAITGEN.JED
CUPL          4.XX  Serial# XX-XXX-XXXX
Device        p16r4  Library DLIB-d-26-11
Created       Thur Aug 20 09:52:02 1990
Name          WaitGen
Partno        P9000183
Revision      02
Date          12/16/89
Designer      Osann
Company       ATI
Assembly      PC Memory;
Location      U106;
*QP20
*QF2048
*G0
*F0
*L00000 11111111111111111111111111111111
*L00032 1011101110111111111111111110111111
*L00256 1011101110111111111111111110111111
*L00288 1111111111111111111111111011111111
*L01024 1111111111111111111111111011111111
*L01056 0111111111111111111111111111111111
*L01088 1111011111111111111111111111111111

```

```
*L01120 11111110111111111111111111111111
*L01152 111111111111111111111111111110111
*L01280 1111111111111111111111101111111
*L01312 01111111111111111111111111111111
*L01344 11110111111111111111111111111111
*L01376 11111110111111111111111111111111
*L01408 11111111111111111101111111111111
*L01536 11111111111111111111111111111111
*L01568 101110110111101101111011111111111
*L01600 101110110111101101111111101111111
*L01792 11111111111111111111111111111111
*L01824 101110110111101110111011111111111
*L01856 101110110111101110111111101111111
*C4D50
*V0001 0XXXXX111N0NННXXXXZHN
*V0002 CXXXXX110N0NНLЛXXZHN
*V0003 000100010N0LНLЛLXXZHN
*V0004 000100100N0LНLЛLXXZHN
*V0005 000101010N0NНLЛLXXZHN
*V0006 000101100N0NНLЛLXXZHN
*V0007 000000100N0NНLЛLXXLLN
*V0008 C00000100N0NНLНХХLLN
*V0009 C00000100N0NННННХХHLN
*V0010 000000110N0NННННХХZHN
*V0011 C00000110N1NНLЛLXXZHN
*3152
```

Шаг 10 — просмотр сигналов после моделирования

Дважды щелкнув левой кнопкой мыши на иконке файла **WAITGEN.SO** можно просмотреть результаты моделирования, которые отображаются на экране как набор сигналов в виде электронной таблицы. Информация по использованию редактора сигналов приведена в подразделе *Просмотр смоделированных временных диаграмм сигналов* раздела *Моделирование проекта на базе ПЛИС*.

Резюме

В этом разделе на простом примере показан переход от концептуального проекта устройства, описанного исходным файлом ПЛИС (или принципиальной схемой) к компиляции проекта, созданию файла тестовых спецификаций и моделированию проекта. Наиболее важными моментами на протяжении описанных этапов были:

- Тщательное рассмотрение задачи и извлечение необходимой информации.
- Выбор микросхемы ПЛИС и распределение выводов.
- Составление промежуточных и логических выражений для описания схемы.
- Запуск компилятора.

- Создание версии проекта, описанного принципиальной схемой.
- Создание и компиляция файла тестовых спецификаций для проверки правильности проекта.
- Запуск программы моделирования.
- Просмотр сигналов, созданных в ходе процесса моделирования.

Примеры проектов на основе ПЛИС

Эта глава содержит примеры проектирования, которые демонстрируют, как используется язык CUPL для описания различных типов проектов. Ниже приведены следующие примеры:

Пример 1 Простые логические элементы (**GATES.PLD**)

Пример 2 Двухразрядный счетчик (**FLOPS.PLD**)

Пример 3 Проект простого конечного автомата (**TURNSTIL.PLD**)

Пример 4 Десятичный реверсивный счетчик, использующий синтаксис конечных автоматов (**COUNT10.PLD**)

Пример 5 Дешифратор для семисегментного индикатора (**HEXDISP.PLD**)

Пример 6 Четырехразрядный счетчик с загрузкой и сбросом

В скобках приведено имя файла описания логики для каждого из этих проектов. Любой из них может быть скомпилирован для создания файла документации или файла загрузки.

Для каждого из файлов описания логики существует соответствующий ему файл тестовых спецификаций с расширением **.SI**, что позволяет запустить программу моделирования для проверки правильности проекта.

Пример 1 — Простые логические элементы

В этом примере детально описано, как написать программу для реализации простых логических элементов, изображенных на рисунке, на базе ПЛИС.

В данном примере присутствует простой набор входных сигналов, а генерируемые выходные сигналы соответствуют результату работы простых логических элементов. Имена выходов назначены с учетом логической функции, выполняемой данным элементом. Например, вентиль И (**AND**) имеет выход, названный **and**.

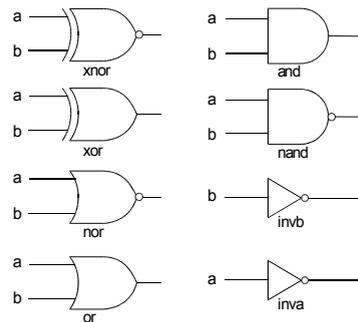


Рис. 5.28. Примеры простых логических элементов.

Ниже приведен исходный код на языке CUPL, описывающий проект **GATES.PLD**.

```

                                GATES.PLD
Name                            Gates;
Partno                           CA0001;
Date                             07/16/87;
Designer                         G Woolheiser;
Company                          ATI;
Location                         San Jose, CA.;
Assembly                         Example

/*****
/*
/* This is an example to demonstrate how the PLD
/* compiler compiles simple gates
/*
/*
/*****
/* Target Devices: P16L8, P16P8, EP300, and 82S153 */
/*****

/* Inputs:  define inputs to build simple gates */
Pin 1 = a;
Pin 2 = b;

/* Outputs:  define outputs as active HI levels

For PAL16L8 and PAL16LD8, De Morgan's Theorem is
applied to invert all outputs due to fixed
inverting buffer in the device.  */

Pin 12 = inva;
Pin 13 = invb;
Pin 14 = and;
Pin 15 = nand;
Pin 16 = or;
Pin 17 = nor;
Pin 18 = xor;
Pin 19 = xnor;

/* Logic:  examples of simple gates expressed in CUPL */

inva = !a;           /* inverters */
invb = !b;
and  = a & b;       /* and gate */
nand = !(a & b);    /* nand gate */
or   = a # b;       /* or gate */
```

```

nor = !(a # b);          /*nor gate      */
xor = a $ b;           /*exclusive or gate */
xnor = !(a $ b);      /*exclusive nor gate */

```

В первой части содержится справочная информация и описание функций, реализуемых в текущем проекте, а также совместимые микросхемы ПЛИС. В первой строке после слова **Name** пишется имя проекта, в дальнейшем компилятор будет использовать его для задания имени выходного файла путем прибавления соответствующего расширения. В строке **Partno** приводится идентификационный код проекта по системе, принятой на данном предприятии. Этот код не имеет никакого отношения к типу целевой микросхемы ПЛИС. В строке **Date** указывается дата компиляции проекта. Для систематизации проектной документации разработчикам рекомендуется изменять дату в этом поле на текущую. В строке **Designer** приводится имя разработчика или группы разработчиков. В строке **Assembly** указывается название или шифр платы, в которой данная ПЛИС будет использоваться (в качестве ключевого слова здесь допускается использовать аббревиатуру **ASSY**). В строке **Location** приводится позиционное обозначение или координаты микросхемы ПЛИС на плате (в качестве ключевого слова здесь допускается использовать аббревиатуру **LOC**). В этой строке может быть представлена и другая информация.

Объявления выводов производятся согласно именам входов и выходов элементов, представленных на рисунке. В данном примере логические элементы имеют два входа — а и b. Далее присваиваются имена выходным выводам. Эти имена выбираются так, чтобы они описывали выполняемую элементом функцию. Использование описательных имен помогает при отладке и последующем использовании проекта.

В секции **Logic** вводятся выражения, которые описывают каждый входящий в проект логический элемент. Для определения состояния выходного вывода как функции от состояний входных выводов а и b используется синтаксис булевых выражений.

Для микросхем PAL16L8 и PAL16LD8, которые содержат фиксированные инвертирующие буферы, компилятор применяет теорему Де Моргана, чтобы проинвертировать все выходы, так как в списке выводов все они объявлены как active-HI, то есть, возбуждаемые высоким уровнем сигнала. Например, компилятор преобразует следующее выражение для элемента ИЛИ (OR), выходной вывод которого объявлен как active-HI:

```
or = a # b ;
```

в следующий совершенный конъюнктивный член (как показано в файле документации):

```
or => !a & !b
```

Микросхемы, выбранные для данного проекта, удовлетворяют всем определенным ранее критериям выбора микросхем — они имеют достаточное количество выводов, как входных, так и выходных; имеют управление на основе использования элементов с тремя состояниями; количество буферизованных и небуферизованных выводов также соответствует требованиям данного проекта и, наконец, эти микросхемы имеют необходимое число комбинационных логических блоков.

Компиляция исходного файла

В этом разделе компилируется файл логических описаний **GATES.PLD**, создаются файлы документации, загрузки в программатор и файл, который в дальнейшем будет использоваться программой моделирования.

Для компиляции файла **GATES.PLD** необходимо выполнить следующие действия:

- В диалоговом окне **Configure PLD** установить следующие опции:

Equations in Doc File	Разрешает генерацию файла документации GATES.DOC , который содержит расширенные логические выражения, символьную таблицу переменных и описание использования комбинационных логических блоков.
Fuse Plot in Doc File	В .DOC файл добавляется карта пережигания перемычек ПЛИС.
Absolute ABS	Создается файл GATES.ABS , используемый в дальнейшем программой моделирования.
Jedec	Генерируется файл GATES.JED , который является входным файлом для программатора, а также используется программой моделирования.

- Для выбора целевой микросхемы нажать кнопку **Change**. В диалоговом окне Target Device выбрать тип микросхемы Device Type 19 и саму микросхему P16L8.
- Нажать **OK**, чтобы закрыть диалоговое окно.
- Сделать файл-источник **GATES.PLD** текущим документом.
- Для запуска процесса компиляции нажать кнопку **Compile**, расположенную на панели инструментов **PldTools**.

Итак, целевая микросхема — PAL16L8, а исходный файл проекта — **GATES.PLD**. Выходными файлами будут **GATES.DOC**, **GATES.ABS** и **GATES.JED**.

После запуска компилятор откроет файл **GATES.DOC**, который содержит сгенерированный расширенный листинг, где показано, как компилятор раскрыл логические выражения при компиляции проекта для выбранной микросхемы.

Для того, чтобы просмотреть отчет об ошибках, сгенерированный компилятором, необходимо:

- Отредактировать исходный файл **GATES.PLD** и удалить символ (;) в конце одного из операторов присваивания, то есть искусственно внести ошибку.
- Сохранить файл.
- Для создания файла списка ошибок нужно в диалоговом окне **Configure PLD** установить флаг формата **Error list LST** во включенное состояние и затем запустить компилятор.

После завершения компиляции в файле **GATES.LST** после каждой ошибочной строки появится строка с сообщением об ошибке, а в начале каждой строки будет проставлен ее номер.

Моделирование проекта

В этом разделе производится моделирование проекта на базе ПЛИС описанного исходным файлом **GATES.PLD**, а затем тестовые векторы добавляются в файл **GATES.JED**, созданный во время компиляции.

Входным файлом для программы моделирования является файл тестовых спецификаций с расширением **.SI**. Для данного примера такой файл, содержащий описание функций устройств, носит название **GATES.SI**. Подробно процесс создания файла тестовых спецификаций описан в подразделе *Создание исходного файла тестовых спецификаций* раздела *Моделирование проекта на базе ПЛИС*.

Программа моделирования сравнивает задающие сигналы на входных выводах и тестовые значения на выходных выводах, введенные в файл **GATES.SI**, как показано ниже, с действительными значениями, вычисленными с помощью логических выражений, описанных в исходном файле на языке CUPL. Эти вычисленные значения помещаются в файл **GATES.ABS**, который создается в процессе компиляции.

Для выполнения моделирования необходимо:

- В оболочке Design Explorer сделать активным документ **GATES.PLD**. Файл **GATES.SI** должен находиться в текущей папке. Не следует пытаться запустить программу моделирования, когда активным документом является файл с расширением **.SI**.
- Нажать кнопку **Simulate**, расположенную на панели управления **PldTools**.

Структурированные тестовые векторы, сгенерированные программой моделирования, будут автоматически добавлены в JEDEC файл, созданный в процессе компиляции.

Входной файл для моделирования **GATES.SI**:

```
Name      Gates;
Partno    000000;
Revision  03;
Date      9/12/83;
Designer  CUPL Engineering;
Company   Protel International.;
Location  None;
Assembly  None;
/*****/
/*
/*      This is a example to demonstrate how the Compiler      */
/*      compiles simple gates.                                  */
/*
/*
/*****/
```

```

/*      Target Devices:  P16L8, P16LD8, P16P8, EP300, and 2S153*/
/*****

/* Order:  define order, polarity, and output      */
/* spacing of stimulus and response values      */

Order:  a, %2, b, %4, inva, %3, invb, %5, and, %8,
        nand, %7, or, %8, nor, %7, xor, %8, xnor;

/* Vectors:  define stimulus and response values, with header*/
/*           and intermediate messages for the simulator listing*/
/* Note:      Don't Care state (X) on inputs is reflected in   */
/*           outputs where appropriate.                        */
Vectors:
$msg " ";
$msg "                               Simple Gates Simulation";
$msg " ";
$msg "  inverters  and   nand   or   nor   xor   xnor
$msg "a b !a !b  a & b !(a & b)  a # b !(a # b)  a $ b !(a $b)";
$msg " - - - - -";

00 HHLHLHLH
01 HLLHHLHL
10 LHLHLHLH
11 LLHLHLHL
1X LXXXHLXX
X1 XLXXHLXX
0X HXLHXXXX
X0 XHLHXXXX
XX XXXXXXXX

```

Ниже приведен выходной файл программы моделирования **GATES.SO**. Входы перечислены рядом с соответствующими выводами.

```

1:Name          Gates;
2:Partno        000000;
3:Revision      03;
4>Date          9/12/83;
5:Designer      CUPL Engineering;
6:Company       Protel International;
7:Location      None;
8:Assembly      None;
9:
10:/******
11:/*

```

```

12:/*      This is a example to demonstrate how the Compiler */
13:/*      compiles simple gates.                               */
14:/*                                                                 */
15:/*-----*/
16:/* Target Devices:P16L8, P16LD8, P16P8, EP300, and 82S153 */
17:/*-----*/
18:
19:
20:/*
21: * Order:  define order, polarity, and output
22: * spacing of stimulus and response values
23: */
24:
25:Order:  a, %2, b, %4, inva, %3, invb, %5, and, %8,
26:        nand, %7, or, %8, nor, %7, xor, %8, xnor;
27:
28:/*
29: * Vectors:  define stimulus and response values, with header
30: *           and intermediate messages for the simulator listing.
31: *
32: * Note: Don't Care state (X) on inputs is reflected in
33: *       outputs where appropriate.
34: */
35:

```

=====
Simulation Results
=====

Simple Gates Simple Simulation										
		inverters		and	nand	or	nor	xor	xnor	
a	a	!a	!b	a&b	!(a&b)	a#b	!(a#b)	a\$b	!(a\$b)	
-	-	-	-	-	-	-	-	-	-	-
0001:	0	0	H	H	L	H	L	H	L	H
0002:	0	1	H	L	L	H	H	L	H	L
0003:	1	0	L	H	L	H	H	L	H	L
0004:	1	1	L	L	H	L	H	L	L	H
0005:	1	X	L	X	X	H	L	X	X	X
0006:	X	1	X	L	X	X	H	L	X	X
0007:	0	X	H	X	L	H	X	X	X	X
0008:	X	0	X	H	L	H	X	X	X	X
0009:	X	X	X	X	X	X	X	X	X	X

Пример 2 — Двухразрядный счетчик

Данный пример демонстрирует реализацию двухразрядного счетчика на D-триггерах. Временные диаграммы сигналов приведены на рисунке 5.29.

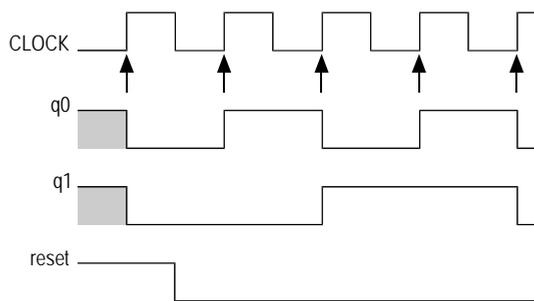


Рис. 5.29. Временные диаграммы работы двухразрядного счетчика.

Как показано стрелками, срабатывание регистров происходит по положительному фронту импульсов синхронизации.

Ниже приведен исходный код на языке CUPL, описывающий проект двухразрядного счетчика (смотри пример **FLOPS.PLD**).

```

Name          FLOPS.PLD
Partno        CA0002;
Revision      02;
Date          07/16/87;
Designer      G. Woolheiser;
Company       ATI;
Location      None;
Assembly      None;
/*****/
/*                                                    */
/* This example demonstrates the use of D-type flip-flop */
/* to implement a two bit counter using the following  */
/* timing diagram.                                     */
/*                                                    */
/* clock  |__| |__| |__| |__| |__| |__| |__| |__| */
/*                                                    */
/* q0     /// |_____| |_____| |_____| |_____| |__ */
/*                                                    */

```

```

/* q1      /// |_____|          |__      */
/*                                     */
/*                                     */
/* reset   |_____|                                     */
/*                                     */
/*****
/*      Target Devices: PAL16R8, PAL16RP8, GAL16V8      */
/*****
Pin 1 = clock;
Pin 2 = reset;

/* Outputs: define outputs and output active levels */
Pin 17 = q0;
Pin 16 = q1;

/* Logic: two bit counter using expanded exclusive ors */
/*      with d-type flip-flop                                     */
q0.d = !reset & (!q0 & !q1
              # !q0 & q1);
q1.d = !reset & (!q0 & q1
              # q0 & !q1);
/* ANDed !reset defines a synchronous register reset */

```

В первой части содержится справочная информация и описание функций, реализуемых в текущем проекте, а также совместимые микросхемы ПЛИС.

Объявления выводов производятся согласно именам входов и выходов элементов, представленных на временной диаграмме.

В секции **Logic** вводятся выражения, описывающие работу счетчика. Выражение для **q0** определяет, когда появляется сигнал **q0**, то есть оно определяет ситуацию непосредственно до прихода переднего фронта тактового сигнала.

Член **!reset**, обеспечивающий синхронный сброс, используется в выражениях, описывающих **q0** и **q1**, для инициализации схемы. При включении питания регистры могут находиться в неопределенном состоянии, что показано символами (/) на временной диаграмме в исходном файле, поэтому сначала выставляется сигнал сброса. Во всех выражениях член **!reset** объединяется элементом И с другими переменными, и после завершения процесса включения питания сигнал **reset** сбрасывается в 0, а член **!reset** соответственно становится равным 1 и перестает влиять на состояние регистров.

Расширение **.d** в выражениях указывает на D-триггер. Однако, когда выход используется как обратная связь, расширение **.d** опускается. Например, если **q0** подается в качестве обратной связи на **q1**, выражение может быть составлено так:

```

q1.d = q0 & !reset ;
но не так:
q1.d = q0.d & !reset ;

```

и не так:

```
q1.d = q0.dq & !reset ;
```

Пример 3 — Проект простого конечного автомата

Контроллер турникета в метро представляет собой простейший пример конечного автомата. Этот контроллер ожидает сигнала, поступающего при опускании жетона. Затем он переходит из состояния "заперто" в состояние "открыто". На этом этапе контроллер ожидает прохода через турникет и, получив такой сигнал, возвращается к закрытому состоянию. Получим граф с двумя вершинами-состояниями — "открыто" и "заперто" и связями-переходами, срабатывающими от детекторов опускания жетона и прохода. На рисунке 5.30 представлена диаграмма такого контроллера, показывающая его состояния и сигналы, инициирующие их изменение.

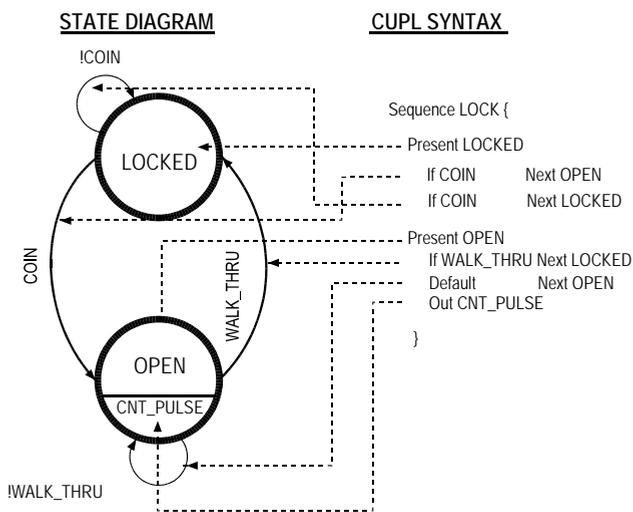


Рис. 5.30. Диаграмма работы конечного автомата.

Справа от диаграммы приведен соответствующий этому конечному автомату код на языке CUPL, а также соответствующие связи для лучшего понимания между операциями и элементами диаграммы.

```

/*****
/*   Target Devices: P16R4
/*****

/* Inputs:   define inputs to Turnstile controller   */
Pin 1 = CLK;
Pin 2 = WALK_THRU;

```

```

Pin 3 = COIN;

/* Outputs:  define outputs as active HI levels
   ..... */
Pin 14 = CNT_PULSE;
Pin 15 = LOCK;

/* Logic:  Subway Turnstile example expressed in CUPL */
$DEFINE LOCKED `b'0
$DEFINE OPEN  `b'1
Sequence LOCK{

Present LOCKED
    if COIN      Next OPEN;
    if !COIN     Next LOCKED;

Present OPEN

    if WALK_THRU Next LOCKED;
    Default      Next OPEN;
    Out CNT_PULSE;
}

```

Пример 4 — Двоично-десятичный реверсивный счетчик

Данный пример описывает четырехразрядный двоично-десятичный счетчик с функцией асинхронного сброса. Счетчик также обеспечивает генерацию асинхронного импульса для управления последующим счетчиком при их каскадном включении. Исходный файл реализует счетчик с использованием синтаксиса CUPL для метода конечных автоматов.

Входной сигнал **DIR** задает направление счета. Когда сигнал **DIR** имеет высокий уровень, то значение счетчика уменьшается на единицу с приходом каждого импульса синхронизации. При низком уровне на входе **DIR** содержимое счетчика, наоборот, увеличивается. Сигнал **CLR** осуществляет синхронный сброс счетчика (рис. 5.31).

Исходный CUPL код, реализующий такой счетчик показан ниже (смотри пример **COUNT10.PLD**).

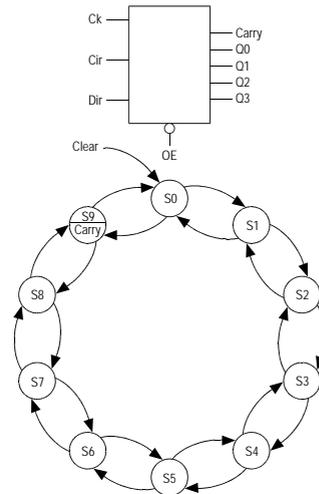


Рис. 5.31. Схема реверсивного десятичного счетчика.

```

COUNT10.PLD
Name          Count10;
Partno        CA0018;
Revision      02;
Date          07/16/87;
Designer      Kahl;
Company       ATI;
Location      None;
Assembly      None;
Device        p16rp4;

/*****
/*
/*          Decade Counter
/* This is a 4-bit up/down decade counter with
/* synchronous clear capability. An asynchronous
/* ripple carry output is provided for cascading
/* multiple devices. CUPL state machine syntax
/* is used
*****/
/* Allowable Target Device Types: PAL16RP4, GAL16V8, EP300 */
*****/
/** Inputs **/
Pin 1 = clk;          /* counter clock          */
Pin 2 = clr;          /* counter clear input    */
Pin 3 = dir;          /* counter direction input*/
Pin 11 = !oe;         /* Register output enable */

/* Outputs
*/

Pin [14..17] = [Q3..0]; /* counter outputs      */
Pin 18 = carry;        /* ripple carry out     */

/* Declarations and Intermediate Variable Definitions */
field count = [Q3..0]; /* declare counter bit field */
#define S0 'b'0000
#define S1 'b'0001
#define S2 'b'0010
#define S3 'b'0011
#define S4 'b'0100
#define S5 'b'0101
#define S6 'b'0110
#define S7 'b'0111
#define S8 'b'1000
#define S9 'b'1001
field node = [clr,dir]; /* declare filed node control */
up = mode:0;           /* define count up mode    */

```

```

down = mode:1;          /* define count down mode    */
clear = mode:[2..3];   /* define count clear mode     */

/* Logic Equations */
sequence count {      /* free running counter      */

present S0           if up           next S1;
                    if down          next S9;
                    if clear         next S0;
present S1           if up           next S2;
                    if down          next S0;
                    if clear         next S0;
present S2           if up           next S3;
                    if down          next S1;
                    if clear         next S0;
present S3           if up           next S4;
                    if down          next S2;
                    if clear         next S0;
present S4           if up           next S5;
                    if down          next S3;
                    if clear         next S0;
present S5           if up           next S6;
                    if down          next S4;
                    if clear         next S0;
present S6           if up           next S7;
                    if down          next S5;
                    if clear         next S0;
present S7           if up           next S8;
                    if down          next S6;
                    if clear         next S0;
present S8           if up           next S9;
                    if down          next S7;
                    if clear         next S0;
present S9           if up           next S0;
                    if down          next S8;
                    if clear         next S0;

out                 carry;          /* assert carry output */

```

Первая часть файла содержит справочную информацию о проекте и описание функции, реализуемой в проекте, а также тип подходящего устройства ПЛИС.

Далее приведено описание входных и выходных выводов в соответствии с диаграммой, приведенной на рисунке.

Секция **Declarations and Intermediate Variable Definitions** файла содержит описание переменных, упрощающих описание схемы.

Имя **count** присвоено выходным переменным **Q3, Q2, Q1 и Q0**.

Команда **\$DEFINE** используется для присвоения имен десяти бинарным состояниям, представленным на выходе системы конечных автоматов. Имена состояний в последствии можно использовать в логических выражениях для представления соответствующих бинарных чисел.

Ключевое слово **FIELD** используется для определения входных сигналов **CLR** и **DIR** в состоянии, названное **mode**. Это состояние описывается следующим набором выражений:

```
up = mode:0;
down = mode:1;
clear = mode [2..3];
```

Состояние **mode** описывает входные сигналы **CLR** и **DIR** таким образом, приведенные выше три выражения становятся эквивалентными следующим выражениям:

```
up = !clr & !dir ;
down = !clr & dir ;
clear = (clr & !dir) # (clr & dir) ;
```

Три различных режима работы счетчика определяются следующим образом:

up	суммирование — на входах DIR и CLR присутствует низкий уровень.
down	вычитание — на входе DIR присутствует высокий уровень, на входе CLR — низкий.
clear	сброс — на входе CLR присутствует высокий уровень, на входе CLR произвольный.

Секция **Logic Equations** файла содержит описание состояний счетчика с помощью синтаксиса конечных автоматов. В первой строке с ключевым словом **SEQUENCE** переменная **count** (то есть выходам **Q3**, **Q2**, **Q1** и **Q0**) присваиваются определенные бинарные значения.

Присутствующие здесь наборы из трех операторов проверки условий предназначены для определения всех возможных состояний выходов на последующем и предыдущем шагах в зависимости от трех режимов работы счетчика. Например, в случае состояния **S4** в режиме суммирования следующим состоянием будет **S5**, в режиме вычитания — **S3**, в режиме сброса счетчик перейдет в состояние **S0**. Данный пример показывает преимущества синтаксиса конечных автоматов для ясного и четкого описания функционирования разрабатываемого устройства.

В данном примере состояние **S0** (бинарное 0000) используется как состояние, в которое устройство переводится при прохождении сигнала **CLR**. Однако, следует помнить о важности начальной установки какого-либо состояния. В общем случае, рекомендуется в качестве начального достоверного состояния использовать состояние с нулевыми значениями на всех выходах, то есть 0000, что не позволит системе "зависать". Это может произойти, если в момент включения напряжения питания на выходах установятся шестнадцатеричные значения А — F, для которых не определено ни одного условного оператора.

Ниже показано, как данный проект может быть записан как виртуальный. Это тот же самый файл, но в нем сделаны изменения, необходимые для демонстрации отличий

между виртуальным проектом и проектом, ориентированным на конкретное устройство.

```

COUNT10.PLD
Name          Count10;
Partno        CA0018;
Revision      02;
Date          07/16/87;
Designer      Kahl;
Company       ATI;
Location      None;
Assembly     None;
Device        VIRTUAL;

/*****
/*
/*          Decade Counter
/* This is a 4-bit up/down decade counter with
/* synchronous clear capability. An asynchronous
/* ripple carry output is provided for cascading
/* multiple devices. CUPL state machine syntax
/* is used
*****/
/* Allowable Target Device Types: PAL16RP4, GAL16V8, EP300 */
*****/
/** Inputs **/
Pin = clk;          /* counter clock          */
Pin = clr;          /* counter clear input    */
Pin = dir;          /* counter direction input */
Pin = !oe;         /* Register output enable */

/** Outputs **/

Pin = [Q3..0];     /* counter outputs        */
Pin = carry;       /* ripple carry out       */

/** Declarations and Intermediate Variable Definitions **/
field count = [Q3..0]; /* declare counter bit field */
#define S0 'b'0000
#define S1 'b'0001
#define S2 'b'0010
#define S3 'b'0011
#define S4 'b'0100
#define S5 'b'0101
#define S6 'b'0110
#define S7 'b'0111

```

```

$define S8 `b'1000
$define S9 `b'1001
field node = [clr,dir];      /* declare filed node control */
up = mode:0;                 /* define count up mode      */
down = mode:1;               /* define count down mode    */
clear = mode:2..3];         /* define count clear mode   */
/* Logic Equations */
sequence count {           /*      free running counter      */

present S0                if up          next S1;
                          if down          next S9;
                          if clear         next S0;
present S1                if up          next S2;
                          if down          next S0;
                          if clear         next S0;
present S2                if up          next S3;
                          if down          next S1;
                          if clear         next S0;
present S3                if up          next S4;
                          if down          next S2;
                          if clear         next S0;
present S4                if up          next S5;
                          if down          next S3;
                          if clear         next S0;
present S5                if up          next S6;
                          if down          next S4;
                          if clear         next S0;
present S6                if up          next S7;
                          if down          next S5;
                          if clear         next S0;
present S7                if up          next S8;
                          if down          next S6;
                          if clear         next S0;
present S8                if up          next S9;
                          if down          next S7;
                          if clear         next S0;
present S9                if up          next S0;
                          if down          next S8;
                          if clear         next S0;
out                        carry;        /* assert carry output */

```

Здесь возможно использовать некоторые функции CUPL препроцессора для значительного сокращения PLD файла. Ниже приведенный пример показывает, как тот же самый файл может быть записан с помощью структуры **\$REPEAT**, позволяющей значительно сократить его объем.

```

Name          Count10;
Partno        CA0018;
Revision      02;
Date          07/16/87;
Designer      Kahl;
Company       ATI;
Location      None;
Assembly      None;
Device        VIRTUAL;

/*****
/*
/*          Decade Counter
/* This is a 4-bit up/down decade counter with
/* synchronous clear capability. An asynchronous
/* ripple carry output is provided for cascading
/* multiple devices. CUPL state machine syntax
/* is used
*****/
/* Allowable Target Device Types: PAL16RP4, GAL16V8, EP300 */
*****/

/** Inputs **/
Pin = clk;          /* counter clock          */
Pin = clr;          /* counter clear input   */
Pin = dir;          /* counter direction input */
Pin = !oe;         /* Register output enable */

/** Outputs **/
Pin = [Q3..0];     /* counter outputs       */
Pin = carry;       /* ripple carry out      */

/* Declarations and Intermediate Variable Definitions */
field count = [Q3..0]; /* declare counter bit field */
field node = [clr,dir]; /* declare filed node control */
up = mode:0;         /* define count up mode    */
down = mode:1;       /* define count down mode  */
clear = mode:2..3]; /* define count clear mode */

/* state machine description */
sequence count {

```

```

present 0
    if up & !clear    next 1;
    if down & !clear next 9;
    if clear          next 0;

$REPEAT i=[1..9]
present i
    if up & !clear    next {(i+1)%10};
    if down & !clear next {(i-1)%10};
    if clear          next 0;
$REPEND
    
```

В этом варианте команда **\$DEFINE** убрана, так как вместо нее используются номера строк. Самым важным изменением здесь является введение циклической конструкции **\$REPEAT**, определяющей большинство логических состояний и используемой, вместо индивидуального их задания. Это стало возможным, так как все состояния идут последовательно и могут быть легко рассчитаны. Исключение составляет состояние 0, определяемое отдельной логической конструкцией.

В цикле для расчета текущего логического состояния используется текущее значение переменной *i*. При работе счетчика в режиме суммирования следующее значение этой переменной определяется как $(i+1)\%10$, что означает суммирование по модулю 10. Это введено, чтобы в состоянии 9 следующее значение $i = 9 + 1 = 10$ по модулю 10 было равно 0. При работе счетчика в режиме вычитания следующее значение этой переменной определяется как $(i-1)\%10$. Так как невозможно выполнить вычитание из 0, введено отдельное определение состояния 0.

Пример 5 — Дешифратор семисегментного жидкокристаллического индикатора

В данном примере рассматривается построение дешифратора, обеспечивающего отображение на семисегментном индикаторе шестнадцатеричного числа, заданного четырехразрядной шиной. В схему включен вход последовательного гашения, предназначенный для гашения незначащих нулей, и выход последовательного гашения для упрощения схемы при каскадировании индикаторов (рис. 5.32).

Сегменты индикатора а — g соединяются с соответствующими выводами устройства, показанного на рисунке.

Ниже приведен исходный код проекта (смотри пример **HEXDISP.PLD**).

```

HEXDISP.PLD
Name           Hexdisp;
Partno         CA0007;
Revision       02;
Date           07/16/87;
    
```

```

Designer          T. Kahl;
Company           ATI;
Location          None;
Assembly          None;
/*****
/*
/*
/*              a              */
/* This is a hexadecimal-to-seven-segment      ---- */
/* decoder capable of driving common-anode      |  | */
/* LEDs. It incorporates both a ripple-        f|  |b */
/* blanking input (to inhibit displaying       |  g  | */
/* leading zeroes) and a ripple blanking output ---- */
/* to allow for easy cascading of digits      |  | */
/*              e|  |c */
/*              |  | */
/*              ---- */
/*              d */
/*
/*****
/* Allowable Target Device Types: 32 x 8 PROM (82S123 or */
/* equivalent
/*****
/** Input group (Note this is only a comment)          **/

pin [10..13] = [D0..3];      /* data input lines to display */
pin 14 = !rbi;              /* ripple blanking input */

/** Output Group ( Note this is only a comment)          **/

pin [7..1] = ![a,b,c,d,e,f,g]; /* Segment output lines */
pin 9 = !rbo;              /* Ripple Blanking output */
/** Declarations and Intermediate Variable Definitions */
field data = [D3..0];      /* hexadecimal input field */
field segment=[abcdefg];   /* Display segment field */
#define ON `b'1           /* segment lit when logically"ON" */
#define OFF `b'0          /* segment dark when logically "OFF" */

```

Первая часть файла содержит справочную информацию о проекте и описание реализуемой в проекте функции, а также тип подходящего устройства ПЛИС.

Далее приведено описание входных и выходных выводов в соответствии с диаграммой, приведенной на рисунке.

Секция файла **Declarations and Intermediate Variable Definitions** содержит определение переменных, упрощающих описание схемы. Группа входных выводов определена как переменная **data**, а выходные выводы названы как **segment**. Значения **ON** и **OFF** соответствуют бинарным значения 1 и 0.

```

/** Logic Equations **/
/*      a      b      c      d      e      f      g */
segment =
/* 0 */      [ ON,  ON,  ON,  ON,  ON,  ON,  ON,  OFF] & data:0 & !rbi
/* 1 */      # [OFF,  ON,  ON,  OFF, OFF, OFF, OFF, OFF] & data:1
/* 2 */      # [ ON,  ON,  OFF, ON,  ON,  OFF, OFF,  ON] & data:2
/* 3 */      # [ ON,  ON,  ON,  ON,  OFF, OFF, OFF,  ON] & data:3
/* 4 */      # [OFF,  ON,  ON,  OFF, OFF, OFF,  ON,  ON] & data:4
/* 5 */      # [ ON,  OFF, ON,  ON,  OFF, ON,  ON,  ON] & data:5
/* 6 */      # [ ON,  OFF, ON,  ON,  ON,  ON,  ON,  ON] & data:6
/* 7 */      # [ ON,  ON,  ON,  OFF, OFF, OFF, OFF,  ON] & data:7
/* 8 */      # [ ON,  ON,  ON,  ON,  ON,  ON,  ON,  OFF] & data:8
/* 9 */      # [ ON,  ON,  ON,  ON,  OFF, ON,  ON,  ON] & data:9
/* A */      # [ ON,  ON,  ON,  OFF, ON,  ON,  ON,  ON] & data:A
/* B */      # [OFF,  OFF, ON,  ON,  ON,  ON,  ON,  ON] & data:B
/* C */      # [ ON,  OFF, OFF, ON,  ON,  ON,  ON,  OFF] & data:C
/* D */      # [OFF,  ON,  ON,  ON,  ON,  OFF, ON,  ON] & data:D
/* E */      # [ ON,  OFF, OFF, ON,  ON,  ON,  ON,  ON] & data:E
/* F */      # [ ON,  OFF, OFF, OFF, ON,  ON,  ON,  ON] & data:F;

rbo = rbi & data:0;

```

Логические выражения здесь задаются таблицей функционирования и описывают, какие сегменты будут загораться при приходе того или иного набора входных данных. Наименования столбцов, соответствующие названиям сегментов, приведены в верхней строке комментариев.

Каждая строка таблицы описывает одно декодируемое шестнадцатеричное значение и набор сегментов, включаемых и выключаемых для его отображения. Например, строка для входного значения 4 будет выглядеть следующим образом:

```
[OFF, ON, ON, OFF, OFF, ON, ON] & data:4
```

В данном примере использование таблицы функционирования позволяет более ясно и четко описать разрабатываемый проект. Из соответствующей строки можно без труда определить, какие сегменты должны быть включены, а какие выключены для отображения того или иного значения.

Пример 6 — Четырехразрядный счетчик с возможностью загрузки и сброса

В данном примере рассматривается построение четырехразрядного счетчика с возможностью загрузки и сброса. Проект построен на базе виртуального устройства, но конечное устройство может быть без труда реализовано почти на любой микросхеме ПЛИС, содержащих четыре и более регистров.

```

Name           Counter;
Partno         FL1201;
Revision       01;
Date          08/26/91;
Designer      RGT;
Company       LDI;
Location      None;
Assembly      None;
Device        VIRTUAL;
/*****/
/* 4-bit counter */
/*****/

/** inputs **/
PIN = clk; /* clock signal for registers */
PIN = load; /* load signal */
PIN = !ClrFlag;
PIN = [LoadPin0..3]; /* pins from which to load data */

/** outputs **/
PIN = [CountPin0..3];

/* intermediate variables and fields */
field STATE_BITS = [Count0..3];
field LOAD_BUS = [LoadPin0..3];

/** state machine definition **/
Sequenced STATE_BITS {
/* build a repeated loop for the states */
$REPEAT i = [0..15]
    Present `h'{i}
        /* go to state 0 if clear signal is true */
        /* if the load signal is false go to the */
        /* next state. Note that the next state */
        /* is (current state + 1) modulo 16. */
        /* This causes the counter to wrap back */
        /* to 0 when in the last state */
If !load Next `h'{(i+1)%16};
If !load & ClrFlag Next `b'0;
$REPEND
/* Add the load capability by using the */
/* APPEND statement. This has the effect of */
/* adding more inputs to the OR gate for */
/* this output. This equation states that if */
/* the load signal is true then the counter */
/* registers are loaded with data from the */
/* load pins. This is why 'load' was used in */

```

```
/* the equations for the 'IF' statements in */
/* the state definitions. */
APPEND STATE_BITS.d = load & LOAD_BUS;
```

Так как это виртуальный проект, нумерация выводов будет игнорироваться, а значит, используемые входные сигналы будут описаны без нумерации выводов.

При появлении сигнала загрузки **LOAD** значение со входа загрузки передается в соответствующий регистр. При появлении сигнала сброса **CLEAR**, все конечные автоматы переводятся в нулевое состояние.

Циклическая конструкция **\$REPEAT** определена в диапазоне [0 ..15]. Внутри цикла текущее состояние определяется как 'h' {i}, что дает возможность обрабатывать числа как шестнадцатеричные. В результате этого реализуется конечный автомат с 16 состояниями, соответствующими шестнадцатеричным числам от 0 до F. Если бы ключевое слово 'h' отсутствовало, то выходные состояния соответствовали бы десятичным значениям от 0 до 15. Компилятор все равно бы принял эти числа за шестнадцатеричные, и значения A — F остались бы не определенными. Кроме того шестнадцатеричное число 10 в данной схеме вообще не может быть реализовано, так как для этого потребуется наличие пятого выходного разряда.

Следует также обратить внимание на то, что во все условные операторы IF добавлено перемножение с переменной **load**, что позволяет придать функции загрузки максимальный приоритет. Кроме того, это позволяет избежать конфликтов, которые могут произойти при приходе сигнала загрузки и некоторых других сигналов одновременно.

Следующее состояние рассчитывается как сумма (текущее значение + 1) по модулю 16, что позволяет перейти в нулевое состояние при достижении последнего значения. Кроме того, использование суммирования по модулю 16 здесь стало возможным, так как число состояний конечного автомата также равно 16.

В последней строке с командой **APPEND** реализована функция загрузки. Эта команда позволяет выполнить операцию **OR** для определенных переменных, что, в свою очередь, дает возможность перевести конечный автомат в состояние, соответствующее набору входных данных. Напомним, что впереди стоит строка с оператором **IF** и переменной **load**, благодаря которым гарантируется отсутствие конфликтов при выполнении команды **APPEND**.

Выражения до команды **APPEND**:

```
CountPin0.d = !load & ???.....
              # !load & ???.....;
```

Выражения после команды **APPEND**:

```
CountPin0.d = !load & ???.....  
             # !load & ???.....  
             # load & LoadPin0;
```

В качестве упражнения, попробуйте реализовать данный счетчик как реверсивный, то есть добавьте в него возможность вычитания. Далее попробуйте реализовать его на базе не виртуального, а конкретного устройства.

Глава 6

Проектирование печатных плат

Редактор печатных плат — основные возможности

Данная глава посвящена работе с редактором чертежей печатных плат PCB Layout Editor и встроенным автотрассировщиком системы Protel 99 SE. Здесь приведена информация, необходимая при работе с системой, описаны основные функции программы по размещению компонентов на плате, выполнению разводки проводников,

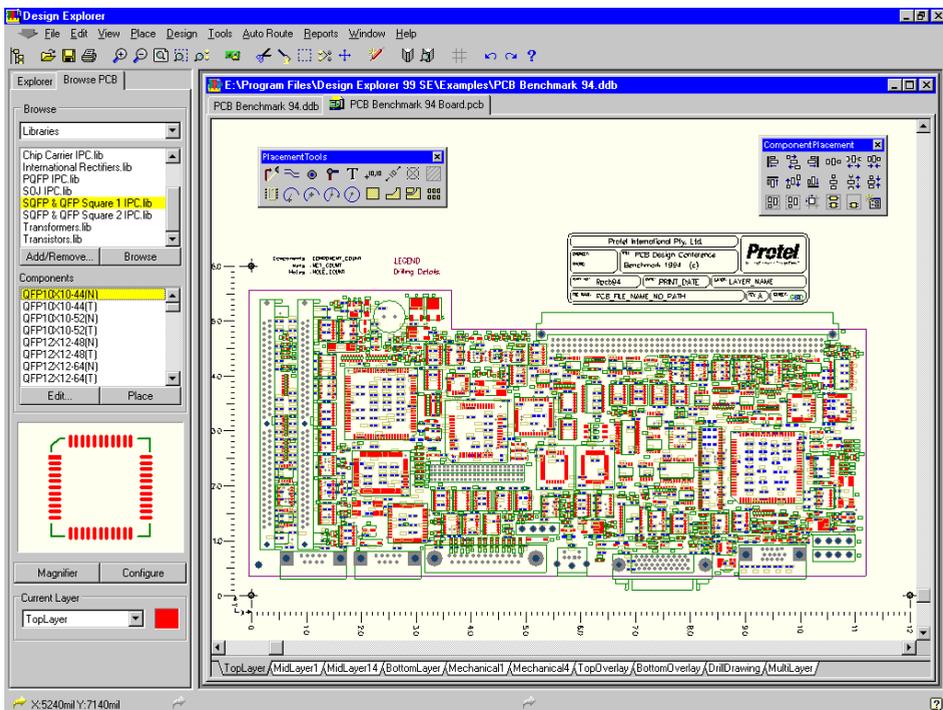


Рис. 6.1. Редактор печатных плат системы Protel 99 SE.

проверке правил электрических соединений, генерации выходных файлов для изготовления фотошаблонов и распечатке конструкторской документации. Более подробную информацию о различных объектах редактора принципиальных схем можно найти в файле электронной справки пакета.

В этой главе также описано, как провести анализ целостности сигналов в цепях только что разработанной платы. Пользователь узнает, как правильно устанавливать правила проектирования для проведения расчета некоторых специфических параметров, таких как импеданс цепей, уровни положительных и отрицательных выбросов импульсов, длительности фронтов и задержек распространения. Кроме того, здесь описывается, как правильно провести моделирование отражений и перекрестных искажений сигналов на разработанной плате, а затем получить истинные временные диаграммы результирующих сигналов в различных проводниках схемы. Подробная информация по всем этим вопросам представлена в разделе *Верификация проекта печатной платы*.

Редактор чертежей печатных плат

Редактор чертежей печатных плат PCB Layout Editor является основным редактором топологических документов системы Protel 99 SE. При помощи этого редактора пользователь может создавать, редактировать и тестировать печатные платы. С помощью редактора печатных плат разработчик может сгенерировать все необходимые выходные файлы, которые будут в дальнейшем использоваться для изготовления фотошаблона печатной платы.

Редактор библиотек топологических посадочных мест

Редактор библиотек топологических посадочных мест PCB Library Editor является вторым редактором топологических документов системы Protel 99 SE. Он используется для создания, редактирования и управления библиотеками топологических посадочных мест и корпусов компонентов. В целом, этот редактор имеет много общего с редактором чертежей печатных плат, но в нем присутствуют ряд дополнительных функций, предназначенных именно для управления библиотеками компонентов.

Бессеточный автотрассировщик

Система Protel 99 SE содержит простое в использовании, но мощное приложение — встроенный в редактор печатных плат бессеточный автотрассировщик. Хотя автотрассировщик подключается к оболочке Design Explorer как отдельный сервер, он настраивается и работает только под управлением редактора печатных плат.

Автотрассировщик системы Protel 99 SE использует разумную комбинацию проверенных временем методов и принципиально новых алгоритмов трассировки проводников. Здесь используются три утилиты **Neural Net**, **Neural Costs** и **Neural Shapes**, базирующиеся на технологии нейронных сетей — основного направления развития искусственного интеллекта. Старые методы трассировки были значительно улучшены, чтобы соответствовать повышенным требованиям современных электронных устройств.

Как и профессиональный конструктор, автотрассировщик решает три основные задачи проектирования: полная разводка проводников, высокая скорость и качество трассировки. Однажды, начав разводить схему при помощи встроенного автотрассировщика системы Protel 99 SE, пользователь будет удивлен тому, как он ранее мог обходиться без него.

Возможности редактора печатных плат

Редактор печатных плат представляет собой полностью законченную систему размещения компонентов и проводников на многослойных платах с множеством полезных функциональных возможностей, обеспечивающих продуктивную работу проектировщика. При использовании редактора печатных плат в комбинации с редактором принципиальных схем, программой моделирования и компилятором устройств на базе ПЛИС он становится полностью автоматизированной, интегрированной и, главное, сквозной системой разработки печатных плат.

Построение топологии печатной платы отличается от других чертежных задач исключительными требованиями к точности прорисовки графических примитивов. Поэтому редактор печатных плат системы Protel 99 SE представляет собой нечто большее, чем просто система прорисовки чертежей.

Другим ключевым отличием является наличие связанности, то есть способности системы распознавать соединения между сегментами проводников, между проводниками и контактными площадками компонентов и так далее. Например, можно удалить сегмент проводника, а система при этом автоматически добавит виртуальную линию связи, таким образом, чтобы всегда обеспечивалась поддержка целостности связей в проекте.

Топология печатной платы создается и отражается на дисплее как набор слоев, которые соответствуют индивидуальным фотошаблонам, используемым при производстве печатной платы, например, верхний и нижний сигнальные слои или слой надписей, выполненных методами шелкографии. Некоторые операции, такие как ручная прокладка проводников, являются зависимыми от выбранного активного слоя.

Независимо от того, что представляет собой проект, будь то простая односторонняя плата или же многослойная плата с множеством внутренних слоев, в редакторе печатных плат системы Protel 99 SE пользователь сможет расположить каждый элемент именно так, как это было задумано изначально.

Синхронизация схемы и печатной платы

Система Protel 99 SE имеет мощный механизм синхронизации различных частей проекта, который позволяет автоматически согласовывать данные на принципиальной схеме с данными на печатной плате. Раньше изменения в схемах переносились на печатные платы с использованием так называемого метода прямого аннотирования, а информация об изменениях в печатных платах возвращалась на этап схемного проектирования посредством метода обратного аннотирования. Теперь обе этих функции выполняет синхронизатор.

В процессе работы синхронизатор сначала проверяет наличие компонентов и связей между ними как на листах принципиальных схем, так и на чертежах печатных плат, а

затем модифицирует их так, чтобы они соответствовали друг другу. При запуске синхронизатора надо выбрать направление синхронизации включением одной из двух опций **Update PCB from Schematic** (обновить плату исходя из изменений в схеме) или **Update Schematic from PCB** (обновить схему исходя из изменений в плате).

32-х разрядная база данных проекта

Редактор использует 32-х разрядную базу данных проекта и может создавать печатные платы как для поверхностного монтажа, так и для монтажа в сквозные отверстия. Система позволяет использовать до шестнадцати сигнальных слоев, четыре дополнительных внутренних слоя питания и заземления, а также четыре механических слоя для прорисовки сборочного чертежа. Максимально допустимый размер платы составляет 100 x 100 дюймов (2540 x 2540 мм). Точность расположения элементов в сетке 0.001 мил (0.0254 мкм) составляет ± 0.0005 мил (± 0.0127 мкм).

Возможность быстрого переключения метрической и дюймовой системы координат позволяет беспрепятственно использовать компоненты с различным шагом выводов.

Широкий набор правил проектирования

Современная электронная промышленность накладывает на расположение компонентов гораздо более широкие требования, чем просто учет их габаритов, а так же их напряжений и токов. Здесь от разработчика требуется применение специфических условий расположения для отдельных цепей, компонентов или областей печатной платы, а также учета таких паразитных факторов, как перекрестные искажения, отражения сигналов и резонансы в длинных линиях.

Для удовлетворения всех этих требований редактор печатных плат содержит в себе широкий набор правил проектирования. Сюда входят контроль зазоров между проводниками, геометрии объектов, параллельности проводников, импеданса, а также приоритеты и топология трассировки. Правила проектирования можно применять ко всей печатной плате, к объектам на ней, цепям, классам цепей, маршрутам, классам маршрутов, компонентам, классам компонентов, слоям или определяемым пользователем областям.

Интерактивная и пакетная проверка правил проектирования

Сообщения обо всех нарушениях правил проектирования встретившихся в процессе разводки платы выводятся разработчику в диалоговом режиме. Пакетная проверка позволяет провести полную проверку компоновки и трассировки печатной платы согласно заданным пользователем физическим и логическим правилам проектирования.

Автоматическое размещение компонентов

Система Protel 99 SE включает две мощные утилиты автоматического размещения компонентов.

Первая из них пользуется для своей работы алгоритмом объединения компонентов в кластеры, который сначала пытается разместить компоненты исходя из информации об их связности, а затем уже с учетом их геометрической формы. Эта утилита полез-

на при проектировании печатных плат с небольшим (менее 100) количеством компонентов.

Вторая утилита применяет так называемый алгоритм модельной "закалки" (Simulated Annealing Algorithm), заимствованный из технологий искусственного интеллекта. Во время размещения компонентов эта утилита, в соответствии с нормами проектирования, анализирует весь проект, учитывая длину соединений, плотность соединений на плате и расположение компонентов. Так как здесь применяются статистические алгоритмы, эта утилита более подходит для проектирования печатных плат с большим (100 и более) количеством компонентов.

Обеспечение целостности связей

Ключевая особенность редактора печатных плат заключается в том, что здесь распознаются и контролируются все логические и физические (или электрические) соединения между элементами. Редактор все время отслеживает состояние целостности связей, добавляя или удаляя виртуальные линии связи при добавлении или удалении отдельных сегментов проводников.

Бессеточная ручная трассировка

Из-за постоянно расширяющегося разнообразия технологий конструктивного исполнения корпусов компонентов современному разработчику сложно проводить трассировку проводников в среде проектирования основанной на фиксированной масштабной сетке. Лучшим решением в этом случае является использование ручной бессеточной трассировки.

Опираясь на удачную комбинацию использования электрической сетки, метода обхода препятствий, шести способов расположения проводников (с предсказанием), а также на собственный опыт, разработчик может правильно прорисовать проводник между любыми объектами.

Гибкие функции выделения элементов

Выделение групп элементов можно производить по слоям, по физической связности или по расположению на печатной плате. К существующему выделению можно добавлять отдельные элементы или, наоборот, удалять таковые. В редакторе печатных плат имеется встроенный мастер запросов, позволяющий строить сложные условия выделения графических примитивов.

С выделенной группой элементов можно производить стандартные манипуляции, например, копирование или удаление в буфер, вставку, зеркальное отображение относительно любой оси и поворот вокруг заданной точки с минимальным угловым шагом 0.001°.

Возможность глобального редактирования

Атрибуты компонентов можно отредактировать в соответствующем диалоговом окне, вызываемом двойным щелчком левой кнопкой мыши на нужном компоненте. В редакторе печатных плат изменения, сделанные для одного объекта, могут быть применены по всему проекту при помощи задания специфических условий. Например,

при редактировании проводника, можно изменить его ширину, его слой, на котором он располагается, или ширину и слой одновременно. Аналогичные изменения пользователь имеет возможность произвести по отношению ко всем проводникам с такой же шириной и/или таким же слоем; для проводников, с шириной отличной от заданной, и/или расположенных на других слоях; ко всем выделенным или невыделенным проводникам. Аналогичные опции глобального редактирования предусмотрены и для других объектов чертежей печатных плат.

Выравнивание компонентов по линии или окружности

В редакторе печатных плат имеется возможность упорядоченного размещения выделенных компонентов по линии или окружности. При размещении компонентов по кругу требуется задать его радиус и угловой шаг. Возможен автоматический поворот компонентов вокруг своей оси.

Операции Undo и Redo

Редактор печатных плат имеет многоуровневый механизм отмены и возврата к состоянию, предшествующему операции отмены. Каждое действие сохраняется в специальной области памяти по стековому принципу. При выполнении команды Undo из стека выбирается последняя операция.

Управление библиотеками компонентов

Во время работы над печатной платой допускается загружать и редактировать несколько библиотек компонентов в редакторе топологических посадочных мест. В стандартной библиотеке системы проектирования содержится более 300 топологических посадочных мест компонентов, предназначенных как для поверхностного монтажа, так и для монтажа в сквозные отверстия. Для сетевой установки в системе поддерживается совместный многопользовательский доступ к библиотекам компонентов.

В редакторе библиотек топологических посадочных мест имеется многофункциональный мастер, автоматизирующий операцию их построения. Процесс управляется рядом диалоговых окон, где пользователь должен включать и выключать различные опции и отвечать на вопросы. В результате можно без труда создавать топологические посадочные места и для простых двухвыводных резисторов, и для микросхем ПЛИС, имеющих сотни выводов, расположенных в несколько рядов на всех четырех сторонах корпуса.

Гибкая система заливки областей металлизации

На любой слой печатной платы можно наложить сплошной или сетчатый металлизированный полигон, с автоматическим подключением его к заданной цепи. Заливка медью будет выполнена с учетом определенных правил проектирования в пределах, очерченных линиями или дугами. Границы заливки можно без труда редактировать в любой момент работы над проектом. После изменения местоположения отдельных компонентов и проводников на плате можно выполнить повторную заливку полигона.

Разделение внутренних слоев питания

Если на плате присутствует несколько шин питания, то они могут быть расположены как на одном, так и на нескольких внутренних слоях питания, причем их разводка будет выполнена с учетом заданных правил проектирования.

Использование тепловых барьеров

В местах присоединения выводов к полигонам или слоям питания или заземления, помимо непосредственного соединения, могут создаваться тепловые барьеры. Пользователь может задавать количество (2 или 4) и ширину проводников барьера, а также зазор между ними.

Управление стеклом контактных площадок на многослойных платах

И для наружных, и для внутренних слоев можно задать контактные площадки различной формы и размеров. При выводе на печать неподключенные контактные площадки, расположенные на внутренних слоях могут быть без труда удалены.

Сквозные и глухие межслойные переходные отверстия

Межслойные соединения на многослойных платах могут быть как сквозные, так и глухие. Сквозные и межслойные переходные отверстия могут размещаться вручную при интерактивной трассировке, индикация соединенных слоев производится с помощью их цвета. Допускается применение сквозных или глухих переходных отверстий на любых двух слоях, поддерживающих технологию наращивания.

Проводники в форме дуг

В редакторе печатных плат на любом из слоев могут быть выполнены проводники в форме дуг с точностью 0.001° , при этом будет производиться контроль связности электрических соединений.

Поворот компонентов

В редакторе печатных плат поддерживается вращение компонентов и их контактных площадок с минимальным шагом 0.001° . Такой же угловой шаг доступен для вращения любого выделенного графического объекта на чертеже платы.

Различные шрифты

Три экранных шрифта (**Default**, **Serif** и **Sans Serif**) обеспечивают качественный вывод текстовой информации на векторные устройства вывода (плоттеры и фотоплоттеры).

Автоматическая генерация фотошаблона

Система Protel 99 SE имеет полностью автоматизированную функцию генерации фотошаблона в формате Gerber 274X для разных слоев, а также файла апертур с возможностью их интерактивного редактирования. Имеется возможность подгонки фотошаблонов к заданным границам на пленке разного размера.

Редактор печатных плат имеет возможность не только импортировать и отображать файлы в формате Gerber, но и загружать наборы таких файлов в заданные слои.

Вывод схем на печать

С помощью одной общей для всех функций печати команды меню Print можно управлять печатью на матричных и лазерных, черно-белых и цветных принтерах, выводом чертежей на плоттер или в формате PostScript, то есть на любое устройство, которое поддерживается операционной системой Windows. Чертежи печатных плат могут выводиться послойно с автоматической центровкой на листе, а также комбинировав на листе изображение нескольких слоев.

Автоматическая генерация файла для сверления отверстий

Файл с таблицей сверления отверстий (NC Drill) создается автоматически без необходимости описания пользовательских массивов данных на инструмент. В генерируемом выходном файле указывается список необходимых инструментов в метрической и дюймовой системе измерения, а также расстояние проходимое каждым инструментом. Для большей эффективности таблицы сверления отверстий сортируются с помощью специального алгоритма.

Редактируемые чертежи сверления отверстий

Чертежи, содержащие карты сверления отверстий могут быть без труда отредактированы пользователем. Рядом с каждым отверстием можно поставить дополнительный маркер: кодовое обозначение (A, B, C и т. д.) или размер отверстия.

Настройки экрана системы Windows

Система Protel 99 SE может работать с цветовым разрешением 24 бит (High Colour) на всех видеокартах и мониторах, поддерживаемых операционной системой Windows. Уровни масштабирования обеспечивают разрешение 32 бита, что соответствует точности ± 0.0005 мил.

Импорт и экспорт файлов в формате DXF

В системе Protel 99 SE имеется возможность импорта и экспорта файлов в формате DXF (стандартном формате системы AutoCAD).

Генерация отчетов

Редактор печатных плат может формировать следующие отчеты: списки используемых материалов (**Bill of Materials**), файлы обратного аннотирования; файлы с таблицами сверления отверстий (**NC Drill**), файлы позиционирования и размещения компонентов (**Pick and Place report**) при монтаже платы; отчеты о статусе списка соединений (**Netlist Status report**); извещения об изменении в конструкторской документации (**Engineering Change Order**) и некоторые другие отчеты.

Настройки редактора чертежей печатных плат

Система координат

Координаты, отображаемые в левой части строки состояния, показывают позицию курсора относительно текущего начала координат чертежа в милах (тысячных долях дюйма) или в миллиметрах, в зависимости от выбранных единиц измерения. Редактор печатных плат позволяет изменять текущее начало координат и располагать его в любом месте чертежа.

Абсолютное начало координат (позиция начала координат по умолчанию) находится в крайнем левом нижнем углу чертежа.

Задание текущего начала координат

Для задания текущего начала координат в текущей позиции курсора необходимо выполнить команду меню **Edit » Origin » Set**. После этой операции установки координаты курсора, отображаемые в строке состояния, сбросятся в ноль и будут показывать **X:0 mils Y:0 mils** или **X:0 мм Y:0мм**, если используется метрическая система мер.

Для возвращения к абсолютному началу координат, расположенному в самом нижнем левом углу чертежа, следует выбрать в меню строку **Edit » Origin » Reset**.

Точность прорисовки объектов

Редактор печатных плат обеспечивает абсолютное разрешение на чертеже в 0.001 mils (0.000001 дюйма или 0.000025 мм), что достаточно для любой практической задачи. Максимально допустимый размер чертежа составляет 100 x 100 дюймов или 2.54 x 2.54 м.

Переключение между дюймовой и метрической системой измерения

Редактор печатных плат системы Protel 99 SE позволяет работать как с дюймовыми (**mils**) так и метрическими (**mm**) единицами измерения. Переключение единиц измерения осуществляется командой меню **View » Toggle Units** или нажатием горячей клавиши **Q** в любой момент работы над проектом. При выборе метрической системы измерения координаты и другая информация о размерах отображаются в миллиметрах (размерность **mm** в строке состояния). Это дает возможность точно задавать размеры платы в мм или создавать новые библиотечные компоненты с метрическим шагом выводов. Для быстрого переключения единиц измерения используется горячая клавиша **Q**.

Система сеток

Редактор печатных плат содержит четыре вида определяемых пользователем сеток. Две первые из них — координатная сетка **Snap Grid** и сетка размещения компонентов **Component Grid** управляют размещением объектов на чертеже. Третья сетка — это электрическая сетка **Electrical Grid**, которая определяет некоторую область вокруг проводника или вывода элемента, при попадании в которую электрические объекты притягиваются друг к другу. Четвертая сетка — видимая сетка **Visible Grid**, облегчающая визуальное ориентирование при перемещении по чертежу.

Snap Grid and Component Grid

Координатная сетка **Snap Grid** представляет собой периодическую структуру точек на чертеже, ограничивающих передвижение курсора и размещение графических примитивов. Сетка **Component Grid** упорядочивает размещение компонентов на чертеже печатной платы. При перемещении указателя мыши можно заметить, что он свободно передвигается между точками координатной сетки. При вызове любой функции редактирования указатель мыши меняет вид и превращается в перекрестие, который будет прыгать только по точкам координатной сетки при работе с примитивами и по точкам сетки размещения при работе с компонентами. При использовании клавиш со стрелками для перемещения по чертежу, курсор всегда будет привязываться к координатной сетке.

Шаг сетки можно изменить в любое время на вкладке **Options** диалогового окна **Document Options**, которое вызывается командой меню **Design » Options**, а также с помощью специальной кнопки **Set Snap Grid** на основной панели инструментов или горячей клавиши **CTRL+G**. Шаг сетки может варьироваться в диапазоне от 0.001 mils до 1000 mils или от 0.000025 до 25.0 мм.

Шаг координатной сетки и сетки размещения компонентов определяет местоположение графических объектов на чертеже, поэтому крайне важно правильно задать его. Шаг сетки обычно выбирается пропорциональным шагом между выводами компонентов. Например, при размещении компонентов с шагом выводов 100 мил необходимо использовать шаг сетки 100 мил или 50 мил. При прокладке проводников между выводами этих компонентов необходимо задать шаг сетки равным 25 мил. Правильное задание шага сетки способствует упорядоченному размещению компонентов и обеспечивает оптимальную трассировку проводников.

Если во время перемещения курсора при помощи клавиатуры удерживать клавишу **SHIFT**, то шаг перемещения увеличивается в 10 раз по сравнению с текущими установками сетки **Snap Grid**.

Электрическая сетка Electrical Grid

Для удобства размещения электротехнических объектов, таких как проводники и переходные отверстия редактор печатных плат использует электрическую сетку. Эта сетка задает некоторую область вокруг электрического объекта, например, проводника, контактной площадки или переходного отверстия, внутри которой перемещаемые объекты будут автоматически притягиваться и сцепляться друг с другом.

Если при перемещении электрического объекта по чертежу он попадает в область электрической сетки другого объекта, то передвигаемый объект зацепится за горячую точку (**Hot Spot**) неподвижного объекта. Электрическая сетка конфигурируется на вкладке **Options** диалогового окна **Document Options**, вызываемого командой меню **Design » Options**.

Электрическая сетка отменяет действие координатной сетки. Это позволяет разработчику легко подсоединиться к любому объекту, находящемуся вне этой сетки. В процессе работы электрическая сетка включается и выключается комбинацией горячих клавиш **SHIFT+E**.

Видимые сетки Visible Grid

Две видимых сетки обеспечивают разработчику визуальный ориентир для размещения и перемещения объектов на чертеже. Сетки могут иметь разный, независимый друг от друга шаг. Например, одна сетка выбирается мелкой, а другая крупной, или одна сетка выбирается метрической, а другая дюймовой.

Видимая сетка отображается на фоне чертежа в виде системы координатных линий или точек. Отображение видимых сеток зависит от текущего уровня масштабирования. Если правильно установленная видимая сетка не показана, то масштаб либо слишком большой, либо слишком маленький.

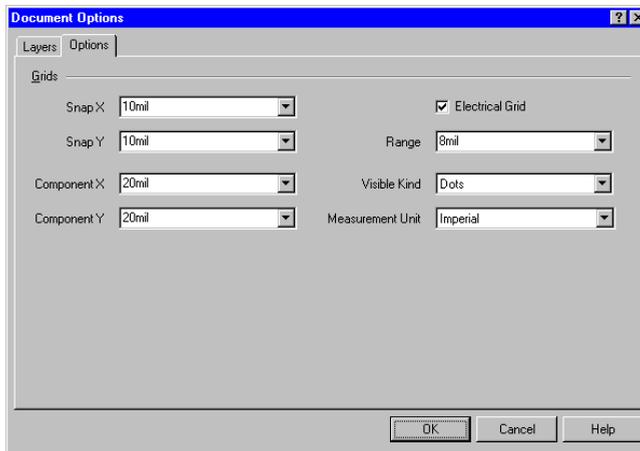


Рис. 6.2. Настройка сеток и выбор системы единиц производится на вкладке Options Tab.

Настройка сеток и единиц измерения

Все сетки настраиваются на вкладке **Options** диалогового окна **Document Options**, вызываемого командой меню **Design » Options**. Здесь также содержатся переключатели единиц измерения и стиля отображения видимых сеток (рис. 6.2).

Работа со слоями

Редактор печатных плат представляет собой многослойную среду разработки печатных плат, в которой пользователь может размещать графические объекты на любом из этих слоев. Слои могут быть либо физическими, из которых берется информация для последующего производства платы, либо системными, как слой соединений, отражающий еще не разведенные связи. К физическим слоям относятся сигнальные слои, внутренние слои питания и заземления, слой надписей, выполняемых методом шелкографии, паяльная маска, маска нанесения паяльной пасты. Для каждого из слоев с помощью команды меню **Tools » Preferences** можно задать свой уникальный идентификационный цвет.

Такая концепция многослойного проектирования отличает редактор печатных плат от многих других прикладных программ черчения или конструирования. Хотя все слои чертежа платы могут быть видны одновременно, с различными слоями могут производиться абсолютно разные операции. Некоторые объекты, например, проводники, полигоны, области металлизации или текстовые надписи должны размещаться на строго заданных слоях.

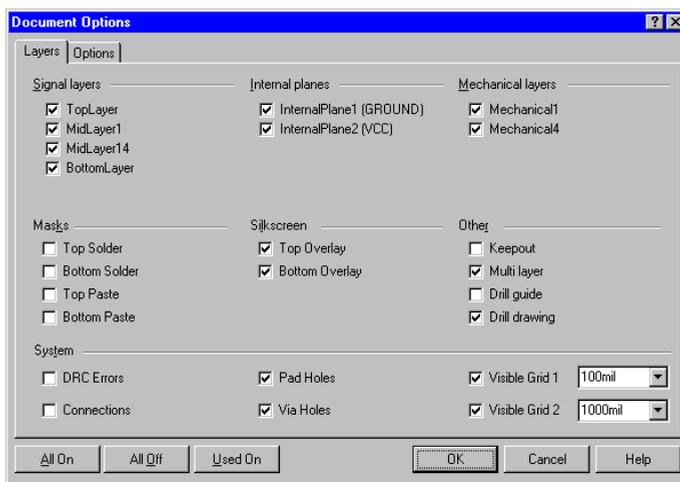


Рис. 6.3. Настройка слоев проекта печатной платы.

Активизация слоев

Для того чтобы разработчик смог получить доступ к любому из слоев, ему необходимо включить этот слой на вкладке **Layers** диалогового окна **Document Options**. После того как слой был включен, вкладка с его именем появится в нижней части окна редактора печатных плат. Подробности по этому вопросу приведены в разделе *Описание печатной платы*.

Для активизации слоев необходимо:

1. Выбрать вкладку **Layers** в диалоговом окне **Document Option**, вызываемом командой меню **Design » Options**.

Обратите внимание, что все слои сгруппированы по типу. Рядом с именем слоя находится переключатель, с помощью которого можно включать и выключать отображение данного слоя. Галочка в окошке переключателя говорит о том, что слой в данный момент активен. Все активизированные слои останутся активными при следующем открытии проекта печатной платы для редактирования.

2. Щелкнуть левой кнопкой мыши на переключателе для активизации требуемого слоя.
3. Щелкнуть по кнопке **ОК**, чтобы закрыть диалоговое окно **Options**.

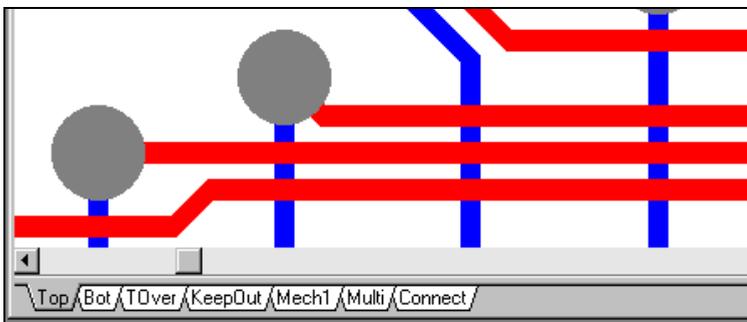


Рис. 6.4. Для каждого активного слоя в нижней части окна редактора печатных плат появляется вкладка с его именем.

Выбор текущего слоя

В любой произвольный момент времени редактор работает в одном из активных слоев чертежа. Такой слой называется текущим, а вкладка с его названием располагается поверх остальных. Некоторые элементы, например, проводники, текстовые надписи, области металлизации и однослойные контактные площадки могут быть размещены только на текущем слое. Другие элементы, такие как компоненты, многослойные контактные площадки и переходные отверстия могут быть размещены на чертеже независимо от выбранного текущего слоя. Операция выделения объектов для последующего перемещения, удаления и т. д. не зависит выбора текущего слоя — пользователь может выполнять эти операции над любыми графическими примитивами без изменения текущего слоя.

Чтобы сделать слой текущим достаточно щелкнуть левой кнопкой мыши на соответствующей вкладке. Клавиши + и – на цифровой клавиатуре используются для последовательного перебора всех активных слоев. Нажатием клавиши **Esc** на цифровой клавиатуре производится последовательный перебор всех активных сигнальных слоев.

Signal Layers

Под трассировку проводников отведено 16 сигнальных слоев. Все, что размещено на этих слоях чертежа печатной платы, будет выполнено в виде медных проводников. Вместе с проводниками на этих слоях могут быть расположены и другие графические примитивы (области металлизации, текстовые надписи, полигоны). Настройка слоев производится в диалоговом окне **Layer Stack Manager**. Сигнальные слои подразделяются на:

- Верхний (**Top**) — сигнальный слой компонентов.
- Внутренние (**Mid Layers**) — сигнальные слои (до 30).
- Нижний (**Bottom**) — сигнальный слой пайки.

Внутренние слои питания и заземления

В распоряжении разработчика имеется 16 сплошных внутренних слоев питания и заземления (пронумерованные как **Copper Plane 1 — 16**), настройка которых производится в диалоговом окне **Layer Stack Manager**. При назначении цепи соответствующего имени они будут автоматически присоединены к этим слоям. К этим слоям в любой момент работы над проектом также могут быть присоединены любые выводы компонентов. При необходимости на данных слоях могут быть образованы контактные площадки с тепловым барьером. Для удобства просмотра на чертежах, выведенных на принтере или плоттере, внутренние слои питания и заземления отображаются в негативе. Иными словами, размещение любого графического примитива на этих плоскостях приведет к образованию там не залитых медью участков (пустот). Имеется возможность разбивать такие слои на несколько отдельных частей и подключать к ним разные цепи. Подробно эта процедура описана в пункте *Создание разделенных слоев питания* раздела *Ручная трассировка печатной платы*.

Слои шелкографии

Слои **Top Overlay** и **Bottom Overlay** предназначены для нанесения рисунков и надписей, выполненных методом шелкографии, верхнюю и нижнюю стороны печатной платы. Здесь обычно отображаются контура корпусов компонентов и их позиционные обозначения, которые автоматически добавляются к ним при вызове и библиотеки. Как правило, контур корпуса компонента создается в редакторе топологических посадочных мест на слое **Top Overlay**. Если компонент размещается на нижнем слое, то чертеж топологического посадочного места со всеми надписями и элементами автоматически зеркально отражается, причем все содержимое слоя **Top Overlay** переносится на слой **Bottom Overlay**.

Механические слои

Для прорисовки вспомогательных элементов чертежа печатной платы, которые не должны быть на саму плату, например, основная надпись, размеры, инструкции по монтажу и сборке, предназначены 16 механических слоев. Все графические примитивы, расположенные на этих слоях могут выводиться на печать одновременно с сигнальными слоями. Настройка механических слоев производится в диалоговом окне **Setup Mechanical Layers**.

Слой масок

Маски для пайки волной

Слои **Top Solder** и **Bottom Solder** предназначены для прорисовки масок для нанесения припоя на верхнюю и нижнюю стороны печатной платы. Эти автоматически генерируемые маски используются при создании трафаретов для пайки волной, обыкновенно прикрывающих всю поверхность платы за исключением выводов компонентов и переходных отверстий. Для удобства просмотра на чертежах, выведенных на принтере или плоттере, эти слои отображаются в негативе. Пользователь имеет возможность управлять параметрами окон в маске посредством установки правила проектирования **Solder Mask Expansion**. Более подробная информация об этом приведена в разделе *Определение требований к проекту печатной платы*. Здесь также описывается порядок маскирования всех переходных отверстий.

Маски для нанесения паяльной пасты

Слои **Top Paste** и **Bottom Paste** предназначены для прорисовки масок для нанесения паяльной пасты на верхнюю и нижнюю стороны печатной платы. Эти автоматически генерируемые маски используются при создании трафаретов для пайки компонентов, использующих технологию поверхностного монтажа. Пользователь имеет возможность управлять параметрами окон в маске посредством установки правила проектирования **Paste Mask Expansion**. Более подробная информация об этом приведена в разделе *Определение требований к проекту печатной платы*. Здесь также описывается порядок маскирования всех переходных отверстий.

Слои для сверления отверстий

Слой сверления отверстий Drill Drawing

Для производства печатной платы обычно используются кодированные чертежи расположения отверстий на печатной плате в слое **Drill Drawing**. Если на плате присутствуют глухие переходные отверстия, то для пар их слоев генерируются отдельные чертежи сверления отверстий. На месте расположения каждого отверстия наносятся позиционирующие символы, сами отверстия обозначаются специальным кодом, рядом проставляется размер. К чертежу можно добавить информацию о количестве отверстий, размерах отверстий в метрической и дюймовой системе измерения, таблице символов. Более подробно эта процедура описана в разделе *Генерация выходных файлов для производства*.

Слой сверления Drill Guide

Здесь приводится чертеж с расположением всех отверстий на печатной плате. Если на плате присутствуют глухие переходные отверстия, то для пар их слоев генерируются отдельные чертежи сверления отверстий. На этих слоях отражаются все контактные площадки и переходные отверстия с размерами отличными от нуля. Более подробно эта процедура описана в разделе *Генерация выходных файлов для производства*.

Прочие слои

Слой **Keep Out**

Этот слой используется для задания области реального размещения компонентов и проводников. Например, границу платы, внутри которой должны быть размещены все компоненты и проводники, можно обозначить сочетанием прямоугольника и дуг. Внутри платы в местах установки крепежных элементов можно задать области, где размещение компонентов и проводников запрещено. Слой **Keep Out** имеет приоритет над всеми сигнальными слоями. Основное правило работы с этим слоем состоит в следующем: ни один компонент не может быть расположен поверх любого объекта, находящегося на слое **Keep Out**, ни один проводник не может проходить через объект, находящегося на слое **Keep Out**.

Слой **Multi Layer**

Объекты, помещенные на этот слой, при формировании выходных файлов отразятся на всех сигнальных слоях. Этот слой обычно используется контактными площадками и переходными отверстиями, имеющими сквозные отверстия.

Системные установки

Connect

Этот переключатель управляет отображением виртуальных линий связи, обозначающих еще не разведенные цепи.

DRC Errors

Этот переключатель управляет отображением маркеров ошибок, возникающих при проверке правил проектирования.

Visible Grid 1 и Visible Grid 2

Эти переключатели управляют отображением двух видимых сеток. Способ отображения сетки (линиями или точками) задается на вкладке **Options**.

Via Holes и Pad Holes

Эти переключатели управляют отображением отверстий контактных площадок и переходных отверстий. Для отличия их друг от друга, используется цвет, устанавливаемый в диалоговом окне **Preferences**.

Настройка редактора печатных плат

Настройка редактора печатных плат производится в диалоговом окне **Preferences**, вызываемом командой меню **Tools » Preferences**, в котором находятся четыре закладки.

Вкладка Options (опции)

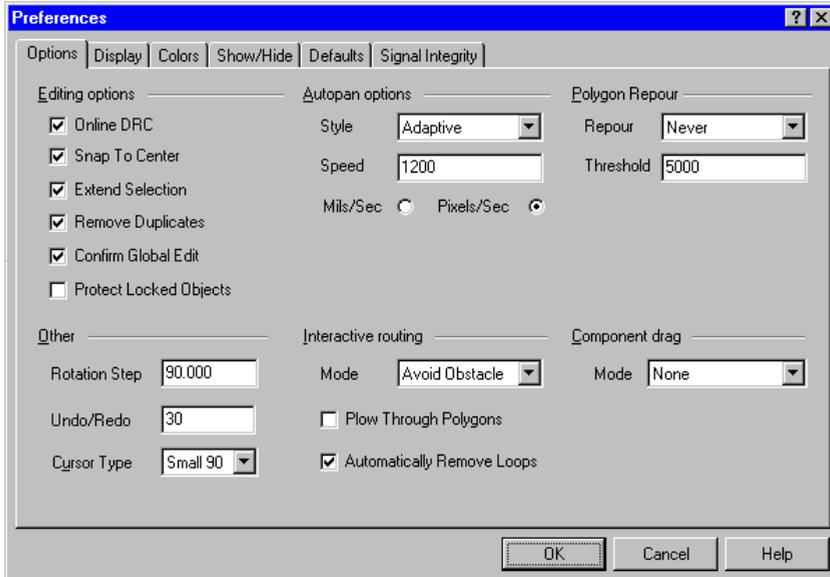


Рис. 6.5. Вкладка Options диалогового окна Preferences.

Поле Editing (редактирование)

Online DRC

Эта опция в режиме реального времени проверяет, не нарушает ли объект, размещаемый в данный момент на чертеже, каких-либо правил проектирования. Правила проектирования задаются в диалоговом окне **Design Rules**, вызываемом командой меню **Design » Rules**. Набор правил, проверяемых в оперативном режиме, задается на вкладке On-line диалогового окна **Design Rules Check**, вызываемого командой **Tools » Design Rules Check**.

Snap to Center

При включении данной опции при выделении и перемещении по чертежу какого-либо объекта, например, контактной площадки или переходного отверстия, указатель мыши захватывает центр площадки или переходного отверстия. При перемещении сегмента проводника указатель мыши захватывает ближайший его конец. При выключенной опции объект будет удерживаться в текущем положении курсора.

Extend Selection

При включении данной опции каждый новый выделяемый объект добавляется к текущему выделению. Если же опция отключена, то при выделении каждого последующего объекта предыдущее выделение снимается.

Remove Duplicates

Включение этой опции инициирует в момент формирования выходных файлов выполнение дополнительной специальной процедуры, проверяющей наличие дублированных примитивов, после чего они будут удалены с чертежа.

Confirm Global Edit

Эта опция предписывает системе при выполнении операции глобального редактирования выводить диалоговое окно, сообщающее о количестве изменяемых объектов и дающее возможность отменить действие в случае возникновения ошибки.

Protect Locked Objects

При включении этой опции заблокированные объекты не могут быть перемещены. Данная опция игнорируется при перемещении группы объектов, некоторые из которых заблокированы.

Другие

Rotation Step

В этом поле задается угол поворота объекта. Захваченный курсором объект поворачивается нажатием клавиши **SPACEBAR** на угол, указанный в поле **Rotation Step**, в направлении против часовой стрелки. При удержании клавиши **SHIFT** и нажатии клавиши **SPACEBAR** объект будет поворачиваться против часовой стрелки.

Undo/Redo

Задаёт максимальное число последовательного выполнения команды отмены предыдущего действия (**Undo**). Следует помнить, что данная команда расходует память, и неоправданно большое число может значительно затруднить работу.

Cursor Type

Это поле позволяет выбрать курсор либо в виде маленького или большого перекрестия, либо в виде маленького перекрестия, повернутого на 45°.

Поле Autopan Options (автоматическое панорамирование)

Style

Эта опция управляет перемещением просматриваемой области чертежа вслед за указателем мыши, когда он имеет вид перекрестия. В выпадающем списке имеется шесть опций:

Re-Center — перемещает центр экрана в место касания курсором края окна. При этом позиция курсора на плате сохраняется.

Fixed Size Jump — перемещает просматриваемую область на расстояние, указанное в поле **Step Size**. При удержании клавиши **SHIFT** размер шага определяется полем **Shift Step Size**.

Shift Accelerate — перемещает просматриваемую область на расстояние, указанное в поле **Step Size**. При удержании клавиши **SHIFT** размер шага плавно изменяется до максимального значения, определяемого полем **Shift Step Size**.

Shift Decelerate — перемещает просматриваемую область на расстояние, указанное в поле **Shift Step Size**. При удержании клавиши **SHIFT** размер шага плавно изменяется до значения, определяемого полем **Step Size**.

Ballistic — скорость перемещения просматриваемой области определяется расстоянием курсора до границы окна редактора при его перемещении за этой границей.

Adaptive — скорость перемещения просматриваемой области задается или в милах в секунду, или в пикселях в секунду.

Step Size

Определяет расстояние, на которое должна переместиться просматриваемая область при касании курсором края окна редактора. Вводимое значение отражается в текущих единицах измерения.

Shift Step Size

Определяет расстояние, на которое должна переместиться просматриваемая область при касании курсором края окна и удержании клавиши **SHIFT**. Вводимое значение отражается в текущих единицах измерения.

Поле Interactive Routing (интерактивная трассировка)

Interactive Routing Mode

Ignore Obstacle — при выборе этого режима объекты можно размещать в любом месте чертежа платы. При включенной опции оперативной проверки плавил проектирования **Online DRC** система незамедлительно выведет сообщение о встреченных ошибках при нарушении установленных зазоров.

Avoid Obstacle — при выборе этого режима пользователю разрешается размещать объекты только там, где они не нарушают установленных зазоров. Эта опция очень полезна при ручной разводке плат, так как она позволяет пользователю размещать объекты в трудных местах платы без боязни нарушить заданные правила проектирования. Более подробная информация по этому вопросу представлена в разделе *Ручная трассировка печатной платы*.

Push Obstacle — при выборе этого режима при прокладке ручной проводников уже размещенные объекты будут расталкиваться согласно ранее заданным проектирования. Если возможности передвинуть мешающий проводник больше нет, то данный режим трассировки сменится на режим **Ignore Obstacle**.

◆ Режим трассировки можно циклически переключать сочетанием клавиш **SHIFT+R**.

Automatically Remove Loops

При включении этой опции система автоматически удаляет с чертежа платы замкнутые петли проводников, которые иногда могут появиться при ручной разводке платы. Рекомендуется включать эту опцию при необходимости повторной трассировки платы.

Plow Through Polygons

Данная опция включает режим прокладывания проводников внутри полигона, после чего заливка полигона выполняется автоматически в зависимости от установок **Re-pour Polygon**.

Component Drag

Эта опция определяет, что делать с проводниками при перемещении компонента. Опция **Enclosed Track** (окруженные проводники) позволяет перемещать вместе с компонентом не только подсоединенные к компоненту проводники, но и проводники, проходящие под ним. Опция **Connected Tracks** (соединенные проводники) позволяет перемещать вместе с компонентом только подсоединенные к нему проводники.

Вкладка Display

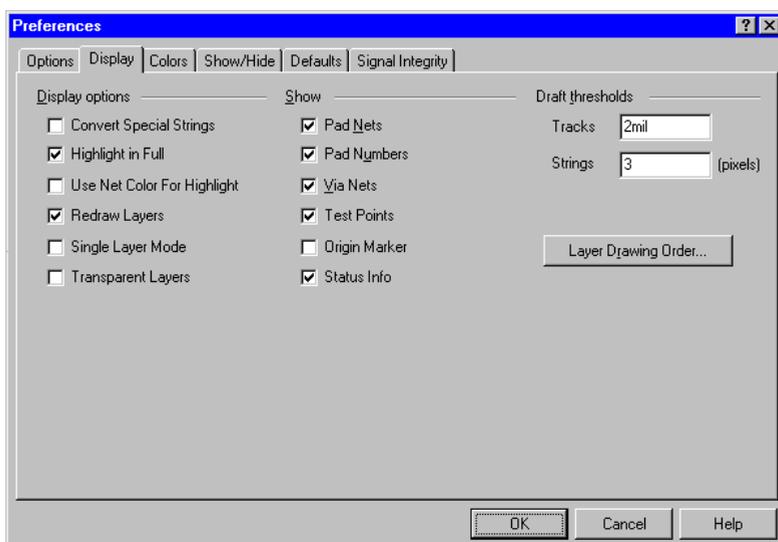


Рис. 6.6. Вкладка Display диалогового окна Preferences.

Поле Display Options

Convert Special Strings

При включении этой опции содержимое таких специальных строк, как **.LAYER_NAME** и **.PRINT_DATE**, отображается на экране всегда, в противном случае на экране отображается имя строки. Подробное описание доступных специальных строк приведено в пункте *Специальные строки* раздела *Объекты редактора печатных плат*.

Highlight in Full

При включении данной опции выделенные объекты целиком подсвечиваются установленным цветом выделения. В противном случае у выделенного объекта подсвечиваются только контуры.

Use Net Color For Highlight

При включении данной опции выделенные цепи подсвечиваются цветом выделенной цепи, задаваемом в диалоговом окне **Change Net**. Для получения большей наглядности рекомендуется использовать опцию **Highlight in Full**.

Redraw Layers

Эта опция управляет перерисовкой слоев при их переключении, причем активный слой будет прорисован последним, то есть поверх остальных.

◆ Для перерисовки только текущего слоя используется комбинация клавиш **ALT+END**.

Single Layer Mode

Эта опция включает режим отображения на экране только текущего слоя. Если активным в настоящий момент выбран сигнальный слой, то отображаются все присутствующие на нем многослойные объекты. Быстрое переключение слоев осуществляется с помощью клавиш + и -, обновление экрана происходит по нажатию клавиши **END**.

Transparent Layers

Эта опция включает режим прозрачности для залитых объектов и позволяет без труда находить объекты, скрытые на другом слое под другим объектом. Она также позволяет без труда идентифицировать объекты, спрятанные другими объектами на текущем слое.

Поле Show

Опции, расположенные в этом поле управляют режимом отображения номеров и имен цепей контактных площадок, имен цепей переходных отверстий, контрольных точек, маркера начала координат и строки состояния.

Поле Draft Threshold

Проводники с толщиной линии равной или меньшей значения, указанного в окне **Tracks**, будут отображаться одной линией, проводники большей ширины будут прорисовываться контуром (в режиме **Draft**).

Строки текста, с высотой строки равной или большей указанного в окне **Strings** значения, будут видны как текст. Если же высота строки составляет всего несколько пикселей, то текст будет выводиться в виде прямоугольника.

Кнопка Layer Drawing Order

Редактор печатных плат позволяет задавать порядок прорисовки слоев на экране дисплея. Для вызова управляющего этим процессом диалогового окна **Layer Drawing Order** следует нажать кнопку **Draw Order**. Здесь приведен список всех слоев согласно

порядку их прорисовки. Слой, находящийся в верхней части списка, отображается на экране дисплея поверх всех остальных слоев.

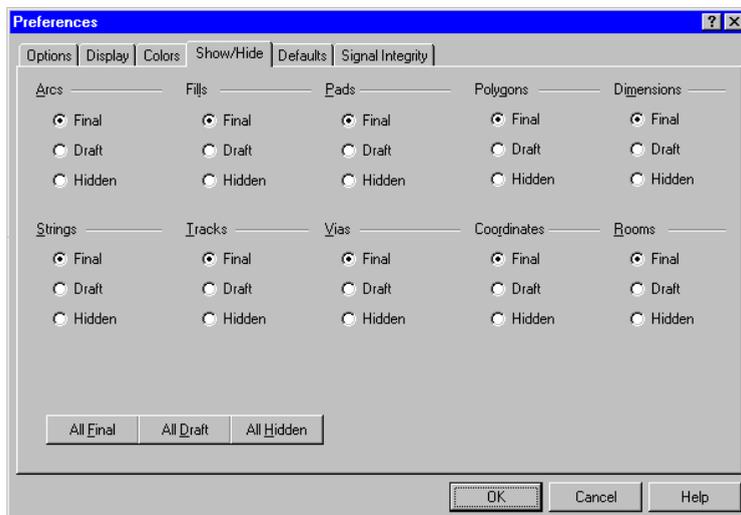


Рис. 6.7. Вкладка Show/Hide диалогового окна Preferences.

Вкладка Show/Hide (просмотр объектов)

Поля Display Mode

Данные поля устанавливают режим отображения для различных типов объектов на чертеже печатной платы. Пользователь может установить один из трех режимов: **Final**, в котором любой объект отображается непрозрачным; **Draft**, в котором отображаются только контуры объектов в соответствии с порогом прорисовки **Draft Threshold**; и **Hidden**, когда объект не прорисовывается вообще. Режим отображения можно установить для каждого типа объекта в отдельности или для всех объектов сразу одновременно с помощью набора кнопок **All**.

Вкладка Default (параметры объектов по умолчанию)

Используемые по умолчанию параметры объектов для данного проекта настраиваются на вкладке **Default** диалогового окна **Preferences**.

Более полная информация об установке атрибутов для каждого типа объектов приведена в разделе *Объекты редактора печатных плат*.

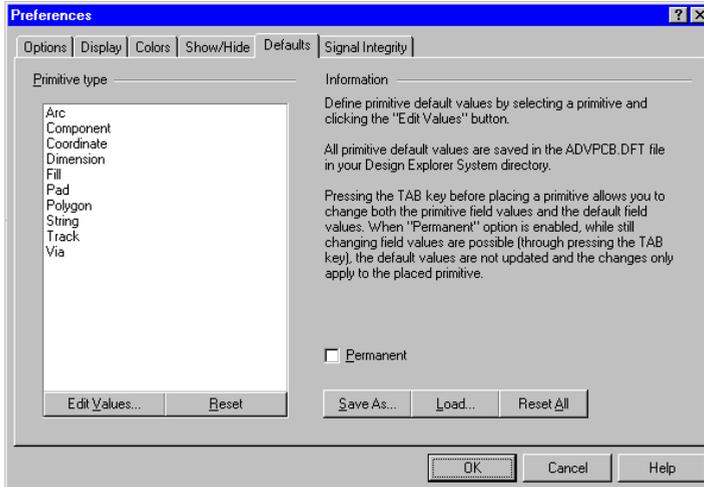


Рис. 6.8. Вкладка Defaults диалогового окна Preferences.

Все установки по умолчанию для определенного типа объектов можно оперативно изменять в любой момент времени работы над проектом посредством нажатия клавиши **TAB**, во время его размещения на чертеже. При включенной опции **Permanent**, расположенной на вкладке **Default**, все оперативно сделанные изменения не затронут установок по умолчанию.

Создание, открытие и сохранение чертежей печатных плат

Основные вопросы по созданию, открытию и сохранению документов были описаны в главе *Работа в среде Design Explorer*. Ниже приводятся только та информация, которая является специфической при работе с чертежами печатных плат.

Создание нового чертежа печатной платы

Редактор печатных плат имеет в своем составе мастер по созданию чертежей печатных плат 1. Он помогает пользователю создать печатную плату на основе готовых шаблонов печатных плат промышленных стандартов или же по своим собственным ранее созданным шаблонам.

Запуск мастера осуществляется командой меню **File » New**. Далее надо перейти на вкладку **Wizards** и выполнить двойной щелчок левой кнопкой мыши на иконке **PCB Wizard**.

Сохранение копии чертежа под новым именем

Сохранить копию активного документа под новым именем или же в текстовом формате Protel можно с помощью команды меню **File » Save Copy As**.

Импорт и экспорт чертежей плат в формате ранних версий системы Protel

Система Protel 99 SE может загружать чертежи печатных плат, созданные ее ранними версиями. Импорт старых проектов может быть осуществлен двумя различными способами.

Можно импортировать чертеж печатной платы как новый документ в существующую базу данных проекта. Подробно эта процедура описана в главе *Работа в среде Design Explorer*. После того, как операция импорта чертежа выполнена, его можно открыть двойным щелчком левой кнопки мыши. Трансляция информации в формат версии Protel 99 SE осуществляется в момент открытия чертежа.

Другой способ заключается в импорте данных о печатной плате в чертеж другой печатной платы. Сделать это можно с помощью команды меню **File » Import**. Далее в поле **Files of Types** надо выбрать нужный старый формат файлов системы Protel и выбрать нужный файл на диске. После нажатия кнопки Open все данные передадутся в активный чертеж.

В системе Protel 99 SE формат чертежей печатных плат был изменен и стал называться PCB 4.0. Для сохранения чертежа в старом формате PCB 3.0, который исполь-

зается версиями Advanced PCB V3, Protel 98, Protel 99 and Protel99 SP1, используется команда меню **File » Save Copy As**.

Работа в редакторе печатных плат

PCB сервер системы Protel 99 SE включает два редактора документов: PCB Layout Editor (редактор топологий печатных плат) и PCB Library Editor (редактор библиотек топологических посадочных мест). Принципы работы в обоих редакторах похожи — это построение проекта из набора доступных объектов, размещение этих объектов в рабочем пространстве. Стратегия, используемая для размещения объектов, редактирования их атрибутов, их позиционирование в и удаление из рабочего пространства и так далее, является общей для обоих редакторов. По существу, способы работы в каждом из этих редакторов одинаковы, разными являются только задачи.

В редакторе библиотек топологических посадочных мест создаются, редактируются и проверяются компоненты и библиотеки компонентов. В редакторе топологий печатных плат создаются, редактируются и проверяются сами печатные платы.

Изменение обзора рабочего пространства

Каждый открываемый пользователем лист чертежа открывается на отдельной вкладке в окне проекта. Панель просмотра проекта и набор команд меню View обеспечивают различные способы просмотра чертежа печатной платы.

Браузер печатных плат MiniViewer

Для оперативного изменения области обзора печатной платы используется браузер MiniViewer, расположенный на панели просмотра редактора печатных плат (рис. 6.9). Прямоугольник из пунктирных линий (**ZoomBox**) показывает поле, отображаемое в окне просмотра активного документа (если плата имеет границу, заданную на слое **Keep Out** или одном механических слоев). Для изменения масштаба просмотра документа в окне проекта необходимо щелкнуть указателем мыши по одному из углов прямоугольника **ZoomBox** и, удерживая левую кнопку мыши, переместить его в нужную точку. Перемещение области просмотра выполняется перетаскиванием масштабирующего прямоугольника в поле браузера с помощью мыши.

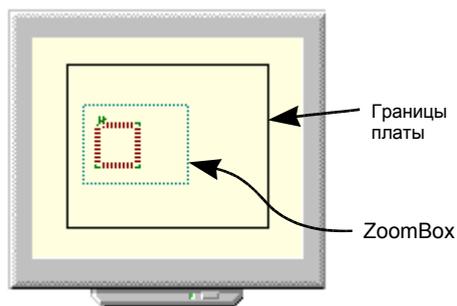


Рис. 6.9. Браузер MiniViewer

Инструмент **Magnifier** (лупа) используется для просмотра в браузере **MiniViewer** участка платы под большим увеличением. С помощью щелчка левой кнопкой мыши по кнопке **Magnifier** и перемещения курсора в виде лупы по плате в главном окне можно подробно рассмотреть интересующую область. Кнопка **Configure** позволяет изменять

степень приближения. Изменять степень увеличения лупы можно перебором трех возможных значений с помощью клавиши **SPACEBAR**.

Команда меню View » Fit Document

Эта команда меняет область просмотра так, что отображаются все объекты, находящиеся в рабочем пространстве чертежа (горячие клавиши: **V, F**).

Команда меню View » Fit Board

Эта команда меняет область просмотра таким образом, что отображается вся плата с учетом границ (горячие клавиши: **V, D**).

Команда меню View » Area

Эта команда переопределяет область просмотра. Новая область задается окном охвата с помощью мыши (горячие клавиши **V, A**).

Команда меню View » Around Point

Эта команда переопределяет область просмотра. Новая область задается окном вокруг задаваемой точки (горячие клавиши **V, P**).

Команды меню View » Zoom

Команда меню **View » Zoom In** приближает область документа, находящуюся вокруг указателя мыши (горячие клавиши **V, I** или **PAGEUP**).

Команда меню **View » Zoom Out** отдаляет область документа, находящуюся вокруг указателя мыши (горячие клавиши **V, O** или **PAGEDOWN**).

Команда меню **View » Zoom Last** возвращает предыдущий вид экрана (горячие клавиши **V, L**). Повторное нажатие **V, L** позволяет переключаться между двумя последними видами экрана.

Для выполнения операций приближения или отдаления с более мелким шагом необходимо при нажатии **PAGEUP** или **PAGEDOWN** удерживать клавишу **SHIFT**.

Перемещение по чертежу печатной платы

Полосы прокрутки

Самым простым средством перемещения по чертежу печатной платы являются полосы прокрутки. Они имеют ползунок, посредством перетаскивания которого можно перемещаться по чертежу во всех направлениях. Положение ползунков указывает на то, какая часть рабочего пространства показывается в данный момент. Прокрутка с большим шагом выполняется с помощью щелчка левой кнопкой мыши по полосе прокрутки, а с маленьким шагом — по стрелкам, расположенным на концах этих полос.

Ручное панорамирование

Для выполнения операции панорамирования рабочего пространства без использования полос прокрутки необходимо выбрать команду меню **View » Pan** (горячие клавиши **V, N** или **HOME**). Эта команда дает возможность выполнить центровку изображения относительно текущего положения указателя мыши. Указатель остается на том же месте, что позволяет продолжить панорамирование.

Операцию панорамирования можно выполнять с помощью клавиш направления. В этом случае каждое нажатие перемещает указатель на один шаг сетки **Snap Grid**. При удержании клавиши **SHIFT** однократное перемещение окна просмотра равно 10 шагам текущей сетки **Snap Grid**.

Автоматическое панорамирование

Автопанорамирование возможно в тех случаях, когда указатель мыши имеет форму перекрестия, то есть при выполнении таких операций редактирования, как размещение, выделение, перемещение или удаление объектов. Такой указатель перемещается либо с помощью мыши, либо посредством клавиш направления. Если при перемещении указатель достигает границы окна, происходит перемещение окна просмотра рабочего пространства. Редактор печатных плат имеет четыре режима, которые выбираются во вкладке Options диалогового окна Preferences.

Быстрый просмотр чертежа печатной платы

Панель управления редактора печатных плат может быть использована для быстрого просмотра содержимого активного чертежа печатной платы. Для этих целей существует шесть режимов просмотра: **Nets** (цепи), **Components** (компоненты), **Net Classes** (классы цепей) и **Component Classes** (классы компонентов), **Rules** (правила проектирования) и **Violations** (предупреждения об их нарушении).

- Если выполняется просмотр в режимах **Nets** или **Components** браузер **MiniViewer** отображает на панели выбранную цепь или компонент (рис. 6.10).
- Для быстрого поиска и идентификации выбранной цепи или компонента в окне главного документа рекомендуется использовать кнопку **Zoom**.
- При выборе элемента из списка информация о нем появляется в строке состояния.

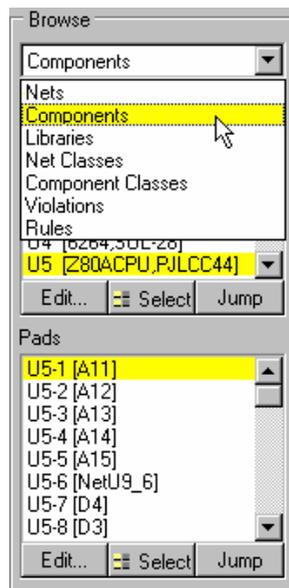


Рис. 6.10. Панель управления редактора печатных плат

Перемещение в заданную точку чертежа пе-

чатной платы

Команда меню **Edit » Jump** (горячая клавиша **J**) позволяет быстро найти и показать нужные компоненты, цепи, контактные площадки компонента, текстовые строки или всю печатную плату без выполнения операций изменения масштаба изображения, панорамирования или прокрутки.

С помощью каждой из этих функций редактор печатных плат переместит указатель мыши на объект и обновит изображение только в том случае, если искомый объект находится за пределами текущей области просмотра. Когда возникает необходимость обновления, объект помещается в центр активного окна.

Опции перемещения включают:

Absolute Origin (абсолютная точка начала координат)

Переход в абсолютное начало координат, то есть в левый нижний угол рабочего пространства (горячие клавиши **J, A** или **CTRL+HOME**).

Current Origin (текущая точка начала координат)

Переход в текущую (относительную) точку начала координат 0,0 (горячие клавиши **J, O** или **CTRL+END**). Эта точка устанавливается с помощью команды меню **Edit » Origin » Set**.

New Location (заданная точка)

Переход к определенной точке чертежа печатной платы. Команда меню **Edit » Jump » New Location** вызывает диалоговое окно **Jump To Location**, в котором текстовые поля **X-Location** и **Y-Location** содержат текущие координаты указателя мыши. При их изменении происходит перемещение в указанное место чертежа (горячие клавиши **J, L**).

Component (компонент)

Переход к указанному компоненту на печатной плате. Команда меню **Edit » Jump » Component** вызывает диалоговое окно **Component Designator**. Необходимо ввести нужное позиционное обозначение и нажать кнопку **OK**. Если название неизвестно, можно набрать символ **?** и нажать клавишу **ENTER** или щелкнуть левой кнопкой мыши, что приведет к сканированию платы и поиску всех размещенных на ней компонентов. Далее нужно выбрать интересующий компонент в диалоговом окне **Components Placed** и нажать кнопку **OK**. Указатель мыши переместится к точке привязки (**reference point**) выбранного компонента (горячие клавиши **J, C**).

Net (цепь)

Переход к ближайшему выводу указанной цепи. Необходимо ввести имя цепи в диалоговом окне **Net Name** и нажать кнопку **OK**. Если название неизвестно, можно набрать символ **?** и нажать клавишу **ENTER** или щелкнуть левой кнопкой мыши, что приведет к сканированию платы и поиску всех цепей. Далее нужно выбрать интересующую цепь в диалоговом окне **Net Loaded** и нажать кнопку **OK**. Указатель мыши переместится к ближайшему выводу, который принадлежит выбранной цепи (горячие клавиши **J, N**).

Pad (контактная площадка)

Переход к заданной контактной площадке указанного компонента. Необходимо ввести имя компонента и номер вывода в диалоговом окне **Jump to Pin Number** (например, **U1-6**) и нажать клавишу **ENTER**. Указатель мыши переместится в центр вывода (горячие клавиши **J, P**).

String (текстовая строка)

Перемещение указателя к поименованной строке. Система выполняет три поисковых операции:

First – Поиск строки, которая совпадает с указанной строкой в обоих регистрах, по всем символам и по длине.

Then – Поиск строки с такими же символами, но которая возможно имеет большую длину.

Finally – Поиск строки с такими же символами, но без учета регистра букв.

Например, при поиске строки "**component**" сначала будет произведен поиск строки "**component**". Если такая строка не найдена, то будет произведен поиск более длинных строк, например, "**components**", а на последнем этапе — таких, как "**CompONENT**". При нахождении строки указатель мыши перемещается к этой строке (горячие клавиши **J, S**).

Error Marker (маркер ошибки)

Выбор этой команды меню приведет к переходу к первому маркеру ошибки. Повторное выполнение — ко второму маркеру ошибки и так далее. Исправленные ошибки удаляются из списка перебора. В противном случае, повтор этой команды ведет к круговому перебору по всем маркерам ошибок в окне активного документа.

Selection (выделение)

Выбор этой команды меню приведет к переходу к первому выбранному объекту. Повторное выполнение — ко второму объекту и так далее. Повтор данной команды ведет к круговому перебору всех выбранных объектов в окне активного документа.

- ◆ Переходы в рабочем пространстве могут быть очень эффективны при работе в редакторе печатных плат, так как это позволяет менять обзор рабочего пространства, не выполняя операции изменения масштаба. Для увеличения скорости все операции перехода можно выполнять с использованием горячих клавиш. Например, для перехода в координаты 1000, 1000 нужно нажать клавиши **J, L**. Появится диалоговое окно **Jump To Location**, в котором будет подсвечено поле **X-Location**. Подсвеченный текст будет заменен на введенный, то есть на 1000. Для перемещения к полю **Y-Location** достаточно нажать клавишу **TAB**. Далее необходимо ввести вторую координату (1000) и нажать клавишу **ENTER**. Диалоговое окно закроется, а указатель мыши переместится в точку с координатами 1000, 1000.

Редактирование объектов на чертежах печатных плат

Существует два подхода к редактированию объектов в редакторе печатных плат — либо посредством изменения их атрибутов в диалоговом окне **Change**, либо графическое модифицирование объектов. Некоторые операции (такие, как изменение цвета примитива) могут быть выполнены только с помощью редактирования атрибутов, другие (такие, как изменение размера заливки) могут быть сделаны обоими способами.

Редактирование внешнего вида объекта, находящегося в рабочем пространстве, проще всего осуществить графическим способом, для чего его необходимо выделить фокусом. Для редактирования объекта посредством диалогового окна **Change** нужно выполнить команду меню **Edit » Change**, а затем щелкнуть на объекте левой кнопкой мыши.

Редактирование во время размещения

Существует возможность редактирования атрибутов объекта во время его размещения на чертеже печатной платы. Когда объект "приклеен" к указателю мыши, необходимо нажать клавишу **TAB**, после чего на экране появится диалоговое окно редактирования атрибутов объекта. У редактирования объектов в процессе размещения имеются следующие преимущества:

- Атрибуты после их изменения могут стать атрибутами по умолчанию для такого типа объектов. Изменения сохраняются в файле по умолчанию **ADVPCB.DFT**. Заметим, что этот метод установки параметров по умолчанию зависит от положения флага **Permanent**, находящегося в диалоговом окне Preferences, вызываемого командой меню **Tools » Preferences**. Если он установлен, тогда изменения не влияют на настройки по умолчанию и применяются только к объекту на указателе.
- Объекты, имеющие числовые идентификаторы, например обозначения контактных площадок, автоматически увеличиваются на единицу.
- Пропадает необходимость в редактировании объекта после его размещения, что ускоряет процесс проектирования.

Изменение уже размещенных объектов

Команда меню **Edit » Change** используется для изменения уже размещенных объектов. Каждый объект имеет свой собственный набор редактируемых атрибутов. Можно выполнять изменения одного объекта или осуществлять расширенные изменения во всем проекте, используя мощные средства глобального редактирования.

Для изменения какого-либо ранее размещенного объекта необходимо выбрать команду меню **Edit » Change**, поместить указатель мыши над объектом и щелкнуть левой кнопкой мыши (двойной щелчок приведет к тому же результату). Если под указателем располагаются несколько объектов, то появится их список, из которого требуется выбрать нужный.

Подробное описание редактируемых атрибутов всех объектов приведено в разделе *Объекты редактора печатных плат*, а в подразделе *Глобальное редактирование* этой же главы обсуждаются вопросы глобального редактирования объектов проекта.

Графическое редактирование — выделение фокусом и комплексное выделение

Одним из основных преимуществ графической среды редактирования является способность внести изменения в объект непосредственно на экране. Начиная процесс графического редактирования отдельного объекта или группы объектов необходимо их выделить. Для этого существует два способа — выделение фокусом и комплексное выделение.

В большинстве приложений операционной системы Windows выделение является единственным способом выбора объектов для их последующего изменения. Типичным примером является выделение одного или более объектов для копирования в буфер обмена и последующей вставки в другое место рабочего пространства. Как правило, модифицируются непосредственно выделенные объекты. Например, в графических редакторах таким образом можно переместить объекты или изменить их размер.

В отличие от других приложений Windows, редактор печатных плат системы Protel использует два независимых метода для решения описанных задач. Оба этих метода, выделение фокусом и комплексное выделение, неоднократно применяются при создании и редактировании чертежа платы. Комплексное выделение позволяет пользователю выполнять более сложные операции модифицирования объектов, которое было бы затруднено или невозможно при использовании метода описанного выше традиционного выделения.

Выделение фокусом

Если в редакторе печатных плат навести курсор на какой-либо объект и щелкнуть на нем левой кнопкой мыши, то объект станет активным, а его вид на экране изменится. Такой метод выделения объекта называется выделение фокусом. Аналогичным образом пользователь делает активным одно из открытых окон приложений Windows.

В любой момент времени выделенным таким образом может быть только один объект, при этом он отображается на экране с особыми графическими элементами — метками-манипуляторами (или фокусным перекрестием). Например, если щелкнуть левой кнопкой мыши по прямоугольнику, в каждом из его углов и в середине каждой из его сторон появится метка, позволяющая изменять его размеры, а в центре — метка вращения. Если проделать такую же процедуру с переходным отверстием, появится фокусное перекрестие. Для выделения фокусом другого объекта необходимо просто щелкнуть по нему левой кнопкой мыши. Сбросить выделение фокусом можно, щелкнув левой кнопкой мыши по пустому месту чертежа печатной платы.

Графическое редактирование объекта, выделенного фокусом

После выделения объекта фокусом можно перемещать его в другое место листа схемы или изменять его графические характеристики.

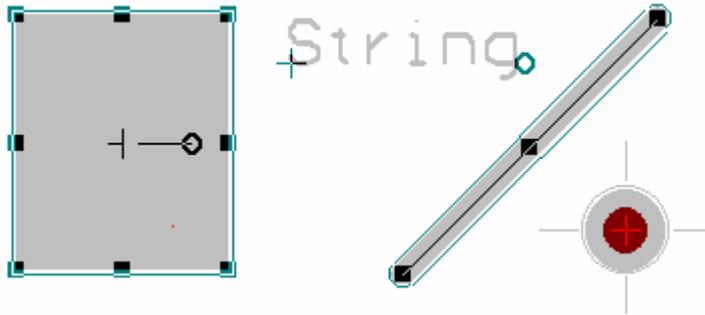


Рис. 6.11. У выделенных фокусом объектов появляются маркеры-манипуляторы

Для того чтобы изменить выделенный фокусом объект непосредственно на листе, нужно щелкнуть по одной из его меток-манипуляторов, после чего эта метка "приклеится" к указателю мыши. Теперь для перемещения объекта на новое место достаточно просто передвинуть туда курсор и повторно щелкнуть левой кнопкой мыши. Для вращения залитого объекта или строки используется метка вращения в виде маленькой окружности.

Если фокусом выделен сегмент проводника, то он имеет три элемента редактирования: два на его концах и один в центре. Первые два используются для перемещения концов проводника, а элемент в центре — для разделения исходного сегмента на две части.

Итак, для перемещения объекта, находящегося в фокусе, нужно щелкнуть по нему мышью, для удаления такого объекта необходимо нажать клавишу **DELETE**. Подробнее методика перемещения объектов описана в подразделе *Перемещение и перетаскивание* данной главы.

Выделение фокусом сегмента проводника

При щелчке мышью по какому-либо сегменту проводника он выделяется фокусом. Если этот сегмент принадлежит некоторой цепи, то все другие объекты, принадлежащие ей, также подсвечиваются, благодаря этому можно легко отследить расположение данной цепи на чертеже печатной платы. Следует помнить, что описываемая функция работает только с теми цепями, которым присвоены имена.

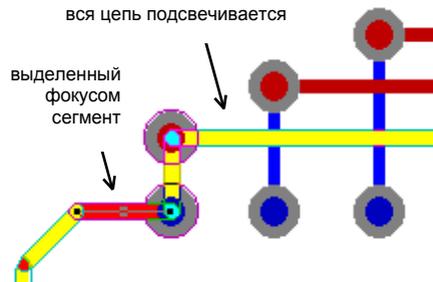


Рис. 6.12. Выделение фокусом сегмента проводника.

Резюме

Как показано в этих примерах, для того, чтобы модифицировать объект непосредственно на чертеже, его необходимо выделить фокусом. Отметим тот факт, что над выделенным фокусом объектом, нельзя совершать операции с применением буфера обмена — копирование, вырезание, вставку, удаление. Эти функции используются только по отношению к комплексно выделенным объектам.

Комплексное выделение

Другим методом манипулирования объектами является комплексное выделение. В отличие от выделения фокусом, этот метод может быть использован по отношению как к отдельным объектам, так к группам объектов.

При комплексном выделении объектов маркеры-манипуляторы не появляются. Вместо этого объект подсвечивается цветом выделения (**selection color**), который задается командой меню **Tools » Preferences**.

Будучи выделенным таким способом, объект может быть перемещен, скопирован, вырезан или вставлен в другое окно или другое место текущего окна, а также удален. Специальные опции команды **Edit » Move** позволяет перемещать или вращать выделенные объекты посредством нажатия "горячих" клавиш. Метод комплексного выделения дает возможность использовать присутствующие в редакторе печатных плат функции глобального редактирования, причем возможно применение их как к выделенным, так и невыбранным объектам (обратное выделение).

◆ Основная особенность метода комплексного выделения в редакторе печатных плат — это возможность выполнения различных операций, не являющихся операциями над выделенными объектами, но требующими использования левой кнопки мыши, без снятия выделения с объектов.

Комплексное выделение в редакторе печатных плат может работать в двух режимах. Оно может быть расширяемым, когда каждая последующая операция выделения добавляет к текущему выделению новые объекты, и они остаются выделенными до тех пор, пока не будет выполнена операция сброса выделения. Другой режим выделения — нерасширяемый. В этом режиме при выполнении очередной операции выделения с помощью команды меню **Edit » Select** происходит сброс предыдущего выделения со всех выделенных ранее объектов. Режим выделения устанавливается во вкладке **Options** диалогового окна **Preferences**.

Выделение производится следующими способами:

- непосредственное выделение, используя клавишу **SHIFT** и левую кнопку мыши для добавления (или удаления) отдельных элементов в (из) текущее выделение;
- с помощью окна охвата;
- командами меню **Edit » Select** и **Edit » DeSelect**;

- с использованием флага **Selection** в диалоговом окне редактирования атрибутов объекта. Эта опция позволяет использовать в редакторе печатных плат функцию глобального редактирования для изменения статуса выделения различных объектов проекта. Подробнее функции глобального редактирования описаны в этой главе в подразделе *Глобальное редактирование*.

- ◆ Выделенные объекты остаются таковыми до тех пор, пока над ними не будет произведена операция снятия выделения с помощью команды меню **Edit » DeSelect** (горячие клавиши — последовательное нажатие клавиш **X, A**).
- ◆ При некорректных действиях пользователь может восстановить предыдущее состояние с помощью команды меню **Edit » Undo**.

Отображение выделенных объектов

Выделенные объекты подсвечиваются цветом, заданным на вкладке **Display** диалогового окна **Preferences**. Существует несколько способов отображения текущего выделения.

Контурный режим (Draft Mode)

Если у графического примитива установлен режим отображения **Draft**, то при выделении он будет обводиться цветом выделения и отображаться в режиме **Final**. Это позволяет легко его идентифицировать. Этот режим отображения устанавливается на вкладке **Show/Hide** диалогового окна **Preferences**.

Полная подсветка (Highlight In Full)

Для закраски всего примитива цветом выделения при его подсвечивании необходимо установить опцию **Highlight In Full** на вкладке **Display** диалогового окна **Preferences**.

Использование для выделения цвета цепи

Цепи могут подсвечиваться индивидуально назначенным цветом, который устанавливается в диалоговом окне **Change Net**, для чего на панели просмотра необходимо установить режим просмотра **Nets** и нажать кнопку **Edit**. Эта опция полезна в сочетании с режимом полной подсветки (**Highlight In Full**). Включение описываемого режима производится с помощью опции **Use Net Color For Highlight**, расположенной на вкладке **Display** диалогового окна **Preferences**.

Выполнение комплексного выделения объектов

Непосредственное выделение отдельных объектов

Непосредственное выделение — наиболее гибкий и удобный способ выделения отдельных объектов. Ниже приведены действия по выделению объекта:

1. Наведите курсор на объект и, удерживая клавишу **SHIFT**, нажмите левую кнопку мыши.

Объект будет обновлен и прорисован цветом выделения (для изменения цвета используйте команду меню **Tools » Preferences**). Повторяя эти действия, можно добавлять другие объекты к текущему выделению.

Если выдается звуковой сигнал или нужный объект не выделяется необходимо приблизить лист схемы (нажать клавишу **PAGEUP**) и убедиться, что указатель мыши находится непосредственно над объектом. При выделении компонента электрической схемы можно щелкнуть в любом месте его условного графического обозначения. Операция выделения компонентов, а особенно составных компонентов, может занять некоторое время.

Для добавления других объектов к текущему выделению:

2. Удерживая клавишу **SHIFT**, нажмите левую кнопку мыши над добавляемым объектом.

Для удаления элемента:

3. Удерживая клавишу **SHIFT**, нажмите левую кнопку мыши на объекте, который нужно удалить из группы выделенных.

После того, как над объектом проведена операция снятия выделения, он перерисовывается своим первоначальным цветом. Выделенные объекты остаются таковыми до тех пор, пока над ними не будет произведена операция снятия выделения либо с помощью одновременного нажатия клавиши **SHIFT** и левой кнопки мыши, либо с помощью команды меню **Edit » DeSelect** (горячие клавиши — последовательное нажатие **X, A**).

<p>◆ Опция Extend Selection, расположенная на вкладке Options в диалоговом окне Preferences определяет расширяемый или не расширяемый режим выделения.</p>

Непосредственное выделение в окне

Для выделения объектов с помощью окна охвата необходимо:

1. Поместить указатель мыши в место, где нет объектов.
2. Нажать и удерживать левую кнопку мыши.

В строке состояния появится подсказка — **Choose Second Corner** (выбери второй угол).

3. Переместить мышь по диагонали, определив прямоугольную область (окно) выделения, и отпустить кнопку мыши.

Будут выделены и перерисованы текущим цветом выделения только те объекты, которые полностью попадают в отмеченный прямоугольник.

4. Для расширения группы выделения нужно повторить все предыдущие действия.

Использование меню для выполнения операции выделения

Команда меню **Edit » Select** позволяет выделить все объекты внутри или снаружи какой-либо области, либо вообще все объекты.

Команды **Select** и **DeSelect** используются для определения сложной группы объектов, которая затем может быть перемещена, скопирована или удалена. Горячая клавиша **S** вызывает всплывающее меню **Select**, а **X** — **DeSelect**.

Опции управления выделением включают:

Inside Area (внутри окна)

Выделенными окажутся только те объекты, которые целиком попадают в окно охвата. Свободные контактные площадки или переходные отверстия включаются в выделение в том случае, если их центр лежит внутри области выделения.

Outside Area (вне окна)

Выделенными окажутся только те объекты, которые не попадают в окно охвата. Правила включения в выделение такие же, как и для **Select** » **Inside Area**. Процедура определения прямоугольного окна охвата также сходна с предыдущим пунктом.

All (все объекты)

Выделение всех объектов на чертеже, включая те, которые имеют скрытое состояние отображения или являются невидимыми из-за того, что выключены слои, в которых они располагаются.

Physical Net (физическая цепь)

Такое выделение будет включать все примитивы (проводники, переходные отверстия, полигоны), которые находятся в физическом контакте с той точкой, где был произведен щелчок мыши, но не будет включать любые части цепи, которые не находятся в физическом контакте, то есть соединяются посредством линий.

Для использования этой функции:

1. Выберите команду меню **Edit** » **Select** » **Physical Net** (горячие клавиши **CTRL+N** или **S, N**).

В строке статуса появится подсказка — **Select Physical Net Starting Point** (выберите начальную точку физической цепи).

2. Расположите указатель мыши над любым примитивом в пределах интересующей цепи и нажмите клавишу **Enter** или левую кнопку мыши.

Непрерывная физическая цепь, которая исходит из точки выделения, будет подсвечена цветом выделения.

All On Layer (все объекты на слое)

Такое выделение будет включать все примитивы, находящиеся в текущем слое, за исключением элементов, которые принадлежат сразу нескольким слоям (обычно это контактные площадки и переходные отверстия).

Free Objects (свободные объекты)

С помощью этой опции в выделение можно включить все объекты, которые не являются частью какой-либо группы (компонент, полигон, размер или координата). Эта функция полезна для возвращения разведенной или частично разведенной платы к

состоянию, предшествующему трассировке. Можно ограничить действие этой функции, выключив слои, содержащие объекты, не требующие выделения.

All Locked (все заблокированные объекты)

В выделение включаются все примитивы и компоненты, у которых включен атрибут **Locked**.

Off Grid Pads

С помощью команды меню **Edit » Select » Off Grid Pads** выделяются все контактные площадки, которые не попадают в текущую сетку **Snap Grid**. Рекомендуется использовать эту опцию перед процессом автотрассировки для контролирования количества контактных площадок, расположенных вне сетки. Для помещения таких компонентов в текущую сетку **Snap Grid** можно использовать команду меню **Tools » Align Components » Move To Grid** (горячие клавиши **A, G**).

Выбор компонента из редактора схем

Для упрощения работы с двумя разными представлениями проекта — принципиальной схемой и чертежом печатной платы в системе Protel 99 SE имеется возможность выделения компонента на плате из выделения на схеме. Для этого необходимо выделить нужный компонент на схеме и выполнить команду **Tools » Select PCB Components** меню редактора схем. Система автоматически переключится в редактор печатных плат, выделит нужный компонент и изменит масштаб просмотра чертежа платы для наилучшего его отображения. Для набора выделенных таким образом компонентов можно создать собственный класс, о чем будет рассказано чуть позже.

Выделение с помощью панели управления

Выделение является наиболее часто применяемой операцией в редакторе печатных плат. Комплексное выделение используется для перемещения, копирования и удаления как отдельных, так и групп компонентов, а также просто для их подсвечивания. Помимо выделения с помощью щелчка левой кнопкой мыши при нажатой клавише **SHIFT** или соответствующей команды меню, пользователь может выделить компонент из панели управления. Здесь можно выделить:

- Текущую цепь или узел этой цепи.
- Текущий компонент или любую его контактную площадку.
- Текущий класс цепей или любую цепь из этого класса.
- Текущий класс компонентов или любой компонент из этого класса.
- Объект по ссылающемуся на него правилу проектирования.

Для добавления нового объекта к текущему выделению, необходимо включить опцию **Set the Extend Selection** в диалоговом окне **Preferences**, в противном случае при выделении нового объекта старое выделение будет автоматически сбрасываться.

Создание класса из текущего выделения

Аналогично комплексному выделению классы представляют собой чрезвычайно удобный инструмент для работы с наборами объектов. Однако, они имеют одно явное преимущество: в отличие от выделения классы можно сохранять и использовать в любой момент работы над проектом, например, для ограничения области действия правил проектирования. Классы могут быть созданы для компонентов, цепей, контактных площадок и маршрутов (**From-To**).

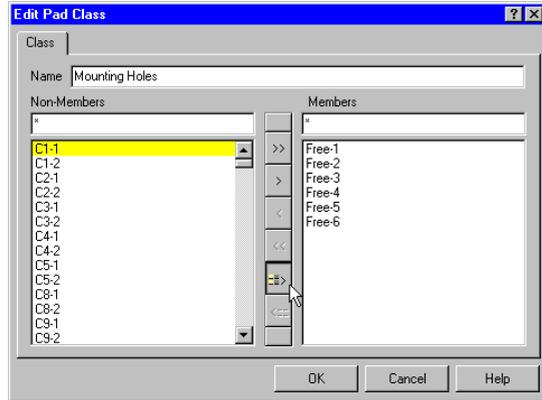


Рис. 6.13. Создание нового класса контактных площадок из набора выделенных

Самый простой способ создания класса объектов заключается в преобразовании текущего комплексного выделения.

Выделение объектов можно выполнить любым из описанных выше стандартных способов, в том числе и с помощью панели управления редактора печатных плат.

Чтобы создать класс из текущего выделения, например, набора контактных площадок, необходимо выполнить команду меню **Design » Classes**. Далее в появившемся диалоговом окне **Object Classes** надо нажать кнопку **Add** на вкладке **Pad**. Затем в диалоговом окне **Edit Pad Class** следует перевести нужные контактные площадки из списка **Non-Members** в список **Members**. Аналогичным образом создаются классы цепей и компонентов.

Последовательный перебор выделенных объектов

Для быстрого нахождения и последовательного перебора объектов из текущего выделения используется панель инструментов **Find Selection**. вызов этой панели осуществляется с помощью команды меню **View » Toolbars » Find Selections** (горячие клавиши **V, B, F**)

Верхний ряд кнопок этой панели инструментов используется для перебора отдельных примитивов выделения, а нижний ряд — для перебора групп примитивов (компонентов, размеров, координат и полигонов).



Рис. 6.14. Панель Find Selections

Сложное выделение по запросу

Запросы для сложных выделений объектов на чертежах печатных плат строятся и сохраняются с помощью модуля **Query Manager**. Менеджер запросов запускается командой меню **Edit » Query Manager**. Для создания нового условия необходимо в окне **Query Manager** ввести его имя в поле **Name**, нажать кнопку **Add**, после чего появится окно **Statement**. Здесь для каждого условия выбирается тип объектов, их параметры и значение для сравнения (равно или не равно). Для сохранения условия необходимо нажать кнопку **OK** и вернуться в окно **Query Manager**. При наличии в запросе нескольких условий они логически перемножаются.

После завершения создания запроса для выполнения выделения следует нажать кнопку **Apply**.

Использование мастера для создания сложных выделений

Редактор печатных плат включает мастер выделения **Selection Wizard**, который можно использовать для создания сложных наборов выделений. Для запуска мастера выделения используется кнопка **Wizard**, расположенная в диалоговом окне **Query Manager**. Этот мастер позволяет выполнить одновременный выбор различных типов примитивов, основанный на некотором наборе определенных пользователем критериев выделения. Например, можно выбрать все контактные площадки и переходные отверстия размером менее 0.5 мм или все проводники, длина которых не равна 8 тысячным дюймам.

Работа с выделенными объектами

Для работы с буфером обмена в редакторе печатных плат используется специальный запатентованный формат, который обеспечивает целостность данных об объектах печатных плат, например, информация о связях на разных слоях. Внутренний буфер обмена пакета Protel не является полностью идентичным стандартному буферу обмена Windows, который позволяет перемещать выделенные объекты, такие, как текстовые строки, между разными приложениями. Графический формат метафайлов (**.WMF**) буфером обмена редактора печатных плат не поддерживается.

Тем не менее, использование буфера обмена в редакторе печатных плат ничем не отличается от других приложений Windows: здесь возможно выделение объектов, вырезание или копирование их в буфер обмена, вставка содержимого буфера в нужное место.

Передачу чертежа платы в другое приложение через буфер обмена системы Windows можно осуществить только через менеджер печати. Подробности по этому вопросу приведены в разделе *Печать чертежа печатной платы средствами Windows* данной главы.

Некоторые замечания по использованию буфера обмена

- Редактор принципиальных схем имеет опцию задания точки привязки (**Clipboard Reference Location**) для операций вырезания или копирования в буфер обмена. При последующей вставке содержимое буфера обмена фиксируется в этой точке, что позволяет правильно и точно расположить вставляемые объекты. Если эта опция включена (команда меню **Tools » Preferences**), то при вы-

полнении операций вырезания или копирования будет выдаваться запрос "**Choose Clipboard Reference Location**" для определения местоположения точки привязки.

- Команда **Edit » Copy** выполняет копирование выделенных объектов в буфер обмена.
- Команда **Edit » Cut** выполняет удаление выделенных объектов с чертежа платы и копирование их в буфер обмена.
- Команда **Edit » Paste** позволяет вставить содержимое буфера обмена в активный чертеж.
- Перед выполнением операций вырезания или копирования необходимо убедиться, что выделенными являются только нужные объекты. Поэтому перед началом выделения необходимо выполнить полную отмену выделения командой **Edit » DeSelect » All** или последовательным нажатием горячих клавиш **X, A**.
- Для добавления объекта к текущему выделению выполните одновременное нажатие клавиши **SHIFT** и левой кнопки мыши.
- Любая операция копирования или вырезания обновляет содержимое буфера обмена, а значит, он будет содержать только последнее выделение.
- Команда **Edit » Clear** удаляет выделенные объекты без помещения их в буфер обмена (горячая клавиша **CTRL+DELETE**).

Специальные опции функции вставки

Специальная вставка позволяет контролировать атрибуты объектов, находящихся в буфере обмена, при вставке их в чертеж печатной платы.

Для управления атрибутами при вставке текущего содержимого буфера обмена необходимо:

1. Выбрать команду меню **Edit » Paste Special**.

Появится диалоговое окно **Paste Special**, которое содержит следующие опции:

Paste on current layer (вставить на текущий слой)

Если эта опция включена, все объекты, которые размещаются в одном слое, такие, как проводники, закрашенные объекты, дуги и контактные площадки, размещающиеся в одном слое, сохраняют свое существующее закрепление за ранее выбранным слоем. При включенной опции все объекты вставляются в текущий слой.

Keep net name (сохранение имени цепи)

Если эта опция активна, все объекты, имеющие имя цепи, сохраняют это имя. При выключении данной опции атрибут цепи устанавливается в значение **No Net**.

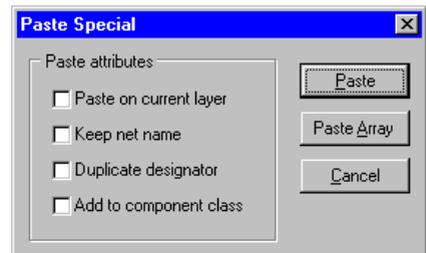


Рис. 6.15. Окно Paste Special

Duplicate designator (дублирование обозначений)

Эта опция поддерживает создание заготовки печатной платы, которая используется там, где нужно скопировать и вставить весь проект. При включении этой опции предыдущая опция сохранения имени цепи, как правило, выключается.

Add to component class (добавление к классу компонентов)

Вставляемые компоненты добавляются в тот же самый класс, что и скопированные компоненты.

2. Нажать кнопку **OK** после настройки опций в диалоговом окне.
3. Разместить выделение в рабочем пространстве и щелкнуть левой кнопкой мыши или нажать клавишу **ENTER**.

Панелизация чертежа печатной платы

Редактор печатных плат может быть использован для создания заготовки печатной платы. С помощью такой заготовки может быть сделано несколько копий одной и той же платы (или разных плат) на чертеже. Для создания заготовки первым делом необходимо скопировать печатную плату в буфер обмена. Затем нужно использовать функцию специальной вставки для копирования печатной платы требуемое число раз. Если имеются трудности с точным позиционированием при операции вставки, можно использовать команду **Jump Location** (горячие клавиши **J, L**), когда вставляемый объект "приклеен" к указателю мыши. Это позволит разместить вставляемый объект без использования мыши. Необходимо помнить, что по полям диалогового окна можно перемещаться с помощью клавиши **TAB** (нажатие **SHIFT+TAB** обеспечивает переход к предыдущему полю), а вместо щелчка по кнопке **OK** — нажимать клавишу **ENTER**.

При панелизации плат с залитыми областями металлизации, система запросит подтверждение на повторную заливку полигонов. В общем случае в этот нет необходимости, так как при панелизации полигоны и объекты рядом с ними не изменялись. Если пользователь выберет операцию обновления полигонов, то система отследит имена подключенных к ним цепей, что может привести к некорректным результатам. Во избежание этого рекомендуется выключать опцию **Keep Net Names** в диалоговом окне **Paste Special**.

Вставка упорядоченных массивов объектов

При использовании **Edit » Cut** (или **Copy**) в буфер обмена помещается копия текущего выделения. Команда **Edit » Paste Array** позволяет разместить в рабочем пространстве несколько копий содержимого буфера обмена.

Для вставки массива текущего содержимого буфера обмена необходимо выбрать команду **Edit » Paste Special**, сделать требуемые настройки в диалоговом окне **Paste Special** и нажать кнопку **Paste Array**. Появится диалоговое окно **Setup Paste Array** (рис. 6.16).

Поле Placement Variables (параметры размещения массива объектов)

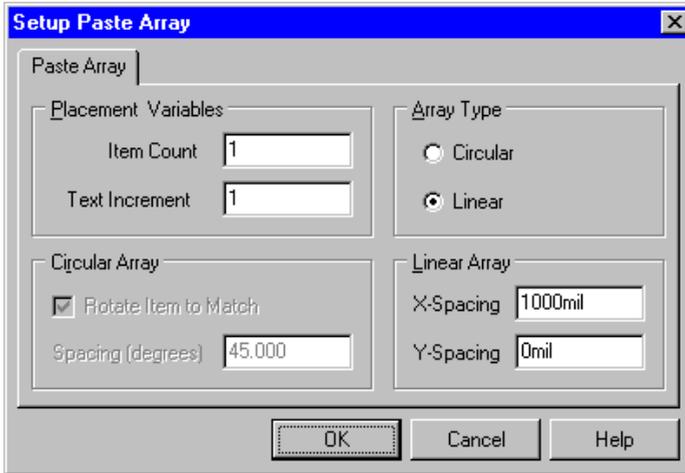


Рис. 6.16. В окне Setup Paste Array задаются параметры размещаемого массива объектов

Item Count

Определяет количество размещаемых копий. Например, ввод в этом окне цифры 4 приведет к размещению четырех копий содержимого буфера обмена.

Text Increment

Данный параметр используется, если нужно, чтобы некоторая составляющая обозначений контактных площадок и компонентов автоматически увеличивалась на заданную величину. Установка в 1 (по умолчанию) приведет к нумерации объектов в наборе, например, **U1**, **U2**, **U3** и так далее.

Поддерживается как численная, так буквенная нумерация. При установке обозначения контактной площадки перед копированием ее в буфер обмена и при определении поля **Text Increment** могут иметь место следующие типы последовательностей обозначений контактных площадок: цифровой (1, 3, 5); буквенный (A, B, C); комбинированный буквенно-цифровой (**A1**, **A2** или **1A**, **1B** или **A1**, **B1** или **1A**, **2A** и так далее).

Для численной нумерации необходимо установить в поле **Text Increment** число, на которое будет выполняться увеличение. Для буквенного инкрементирования в этом же поле нужно установить букву, определяющую число букв, которое нужно пропускать при инкрементировании. Например, если исходная контактная площадка имела обозначение **1A**, а поле в **Text Increment** была установлена буква **C** (третья буква алфавита), копируемые контактные площадки будут иметь такие обозначения: **1A**, **1D** (три буквы после A), **1G** (три буквы после D) и так далее.

Поле Array Type (типы массивов)

Circular (круговой)

В этом случае объекты размещаются по кругу с заданным угловым шагом, задаваемым в поле **Circular Array**.

Linear (линейный)

В этом случае объекты размещаются в узлы прямоугольной сетки с шагом по оси X и Y, задаваемым в поле **Linear Array**.

Поле Circular Array (круговой массив)

Rotate Item to match (вращение элемента)

Элементы массива поворачиваются на угол, установленный в параметре **Spacing**.

Spacing (шаг)

Определяет угловой шаг между вставляемыми элементами. Редактор печатных плат имеет угловое разрешение, равное 0.001°.

Поле Linear Array (линейный массив)

Здесь задается шаг прямоугольной сетки размещения элементов массива.

Введите нужные значения и нажмите **OK**.

После этого в строке состояния появится одна из двух подсказок:

Select Starting Point For Array (введите начальную точку массива) при размещении линейного массива или

Select Center Of Circular Array (введите центральную точку кругового массива) при размещении кругового массива.

Расположите указатель в нужном месте и нажмите клавишу **ENTER** или левую кнопку мыши.

Глобальное редактирование

Редактор печатных плат позволяет редактировать атрибуты не только отдельного объекта, но и всех объектов аналогичного типа в составе выбранного компонента или во всем активном документе.

Пользователь может задавать признаки, по которым будут выбираться объекты для глобального редактирования. Например, изменениям могут быть подвергнуты только те объекты, которые входят в текущее выделение, либо, наоборот, только те, которые не входят в него, либо все объекты, невзирая на их статус выделения. Если это необходимо, имеется возможность создать комплексный набор признаков для применения функции глобального редактирования.

Фактически любой из атрибутов компонентов может быть задействован в той или иной ситуации для глобального редактирования. Самым простым примером является изменение размера всех контактных площадок, связанных с указанным компонентом. Другой пример — изменение ширины проводников определенной цепи. При глобальном редактировании могут быть использованы любые атрибуты объектов, возможные применения метода ограничены только воображением разработчика.

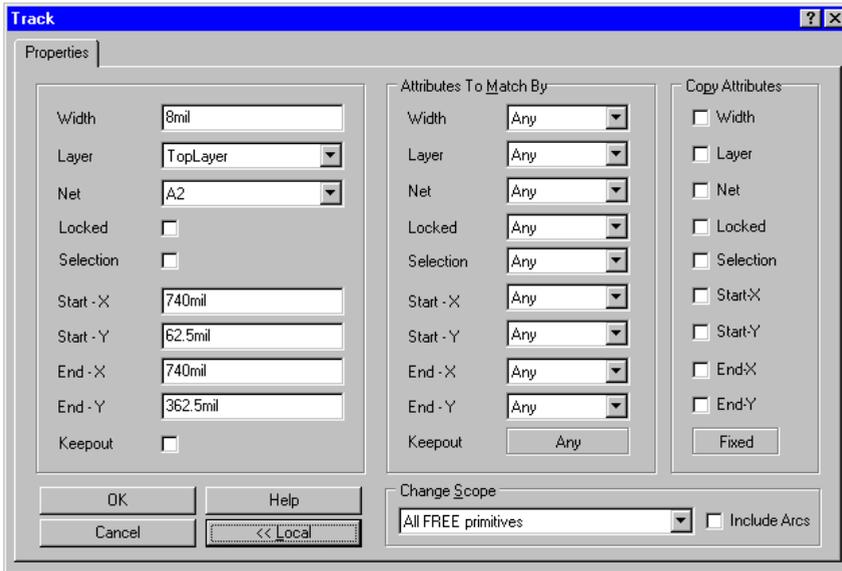


Рис. 6.17. При нажатии кнопки Global в окне появится три столбца параметров.

Диалоговые окна редактирования разных объектов могут содержать разные настройки, так как каждый тип объектов имеет уникальный для него набор атрибутов. Большое количество опций глобального редактирования могут сначала показаться излишне сложными и запутанными. Однако, научившись их эффективно применять, пользователь получит в распоряжение незаменимый инструмент, который избавит его от большого объема работ по ручному редактированию объектов при разработке печатной платы.

Стратегия глобального редактирования

Если набор признаков глобального редактирования, которые присутствуют в диалоговых окнах разных объектов, зависит от этих объектов, то используемая стратегия всегда неизменна. Это описание обрисовывает общий подход к глобальному редактированию.

Current Attributes (текущие атрибуты)

С помощью двойного щелчка левой кнопкой мыши по объекту вызывается диалоговое окно редактирования объектов этого типа. Это окно содержит текущие значения или установки атрибутов данного объекта, которые, разумеется, можно изменять.

При нажатии на кнопку **Global** произойдет расширение этого диалогового окна, и оно окажется разделено на три области: **Attributes**, **Attributes to Match By** и **Copy Attributes**.

Attributes to Match By

В центре окна располагается столбец с заголовком **Attributes to Match By**. В этом столбце определяются признаки идентификации объектов, к которым нужно применить функцию глобального редактирования.

Столбец **Attributes to Match By** содержит либо выпадающие списки, либо текстовые поля, в которые можно ввести текст с клавиатуры.

Выпадающие списки имеют три состояния: **Same** (применить глобальное изменение к объектам, у которых значение этого атрибута совпадает с текущим), **Different** (применить глобальное изменение к объектам, у которых значение этого атрибута не совпадает с текущим) и **Any** (состояние этого атрибута не влияет на выбор объекта при глобальном изменении). Состояние **Any** является состоянием по умолчанию.

Комбинация этих атрибутов определяет индивидуальный набор объектов, к которому будут применены глобальные изменения.

Если все атрибуты в этом столбце установлены в **Any**, а текстовые поля содержат символ-заменитель *, то глобальные изменения будут применены ко всем без исключения объектам данного типа.

Copy Attributes

Третий столбец в диалоговом окне называется **Copy Attributes**. Этот столбец содержит либо поля флагов для каждого атрибута, либо текстовое поле, в которое можно ввести текст с клавиатуры.

В этом столбце определяется, какие атрибуты нужно изменить и, если присутствует текстовое поле, как его значение должно измениться в соответствующих объектах.

Change Scope (Изменение Области Действия)

Последний устанавливаемый в этом окне параметр задается выпадающим списком **Change Scope** (изменение области действия). Он имеет два состояния: **All primitives** (все примитивы) или **All FREE primitives** (все свободные примитивы). То есть редактор печатных плат имеет возможность идентифицировать все примитивы, которые не являются частью сложного составного объекта, как свободные.

Примеры использования функции глобального редактирования

Приведенные ниже примеры дают некоторое представление о потенциальных возможностях функции глобального редактирования для выполнения изменений параметров компонентов и примитивов.

Пример 1 — Перенос проводников на другой слой

Для перемещения всех проводников с верхнего слоя (**Top**) на нижний (**Bottom**) без учета ширины проводника или его статуса выделения необходимо:

1. Выполнить двойной щелчок левой кнопкой мыши на любом проводнике, расположенном на верхнем слое, после чего откроется диалоговое окно **Change Track**.
2. Установить атрибут **Layer** в значение **Bottom Layer**.
3. Нажать кнопку **Global** для просмотра параметров глобального редактирования.
4. В столбце **Attributes to Match By** установить поле **Layer** в значение **Same**. Другие атрибуты должны иметь значение **Any**.

Сделанные установки заставят редактор печатных плат распространить изменения на все проводники, лежащие в том же слое. Заметим, что эта функция будет учитывать, какой слой был у проводника, а не тот, который будет.

5. В столбце **Copy Attributes** атрибут **Layer** будет автоматически активизирован.

Это заставит редактор печатных плат скопировать изменения, сделанные над атрибутом **Layer**, во все проводники, которые удовлетворяют заданному критерию.

6. Установить параметр **Change Scope** в значение **All FREE primitives**. Если трассировка включает дуги, тогда нужно включить опцию **Include Arcs**, после чего нажать кнопку **OK** или клавишу **ENTER**.

Первым на нижний слой будет перенесен начальный проводник, который подвергся редактированию. Затем появится диалоговое окно для подтверждения глобальных изменений.

7. Для подтверждения сделанных изменений нажать **YES**.

Все проводники, лежащие в верхнем слое, будут перенесены в нижний слой. Для обновления содержимого экрана следует нажать клавишу **END**.

Если нужно перенести какую-либо цепь на другой слой, необходимо выполнить команду меню **Edit » Select » Physical Net** и щелкнуть в любом месте интересующей цепи. Далее нужно повторить описанный процесс глобального редактирования, но в столбце **Attributes to Match By** установить атрибут **Selection** в значение **Same**, а другие атрибуты в значение **Any**. В этом случае на нижний слой переместится только выделенная цепь.

Пример 2 — Изменение диаметра переходных отверстий

Для установки диаметра всех переходных отверстий на плате в значение 0.040 дюйма необходимо:

1. Выполнить двойной щелчок по любому переходному отверстию, после чего откроется диалоговое окно **Change Via**.
2. В поле **Diameter** в столбце **Attributes** ввести значение **40**.
3. Нажать кнопку **Global**.
4. Установить все атрибуты в столбце **Attributes to Match By** в значение **Any** (то есть будут изменены все переходные отверстия).
5. В столбце **Copy Attributes** атрибут **Diameter** будет автоматически активизирован.

Другие атрибуты следует оставить выключенными.

6. Установить параметр **Change Scope** в значение **All primitives** и нажать кнопку **OK** или клавишу **ENTER**.

Выбранное первым переходное отверстие будет перерисовано с новым диаметром, а затем появится диалоговое окно для подтверждения глобальных изменений.

7. Нажать кнопку **OK**.

После этого изменится диаметр всех переходных отверстий активного документа.

Пример 3 — Блокировка цепи

Для предотвращения изменения программой автотрассировки некоторой цепи, которая предварительно была прорисована вручную, ее можно заблокировать. Для этого необходимо:

1. Выбрать цепь, то есть выполнить команду меню **Edit » Select » Physical Net** и щелкнуть в любом месте этой цепи.

Все сегменты проводника, которые составляют выбранную цепь, будут подсвечены цветом выделения.

2. Осуществить двойной щелчок левой кнопкой мыши на одном из сегментов выбранного проводника.

Появится диалоговое окно **Change Track**.

3. Включить атрибут **Locked** в диалоговом окне **Change Track**.

4. Нажать кнопку **Global**.

5. Установить атрибут **Selection** в столбце **Attributes to Match By** в значение **Same**.

6. Установить параметр **Change Scope** в значение **All FREE primitives**. Если трассировка включает дуги, тогда нужно включить опцию **Include Arcs**.

Глобальные изменения затронут все выбранные сегменты проводника, не являющиеся частью группового объекта.

7. В столбце **Copy Attributes** атрибут **Locked** будет автоматически активизирован.

Это заставит редактор печатных плат скопировать изменения, сделанные над атрибутом **Locked** выбранного сегмента проводника, во все сегменты, которые удовлетворяют заданному критерию.

8. Нажать кнопку **OK**.

Выбранный первым проводник будет изменен сразу, а затем появится диалоговое окно для подтверждения глобальных изменений.

9. Чтобы заблокировать все остальные выбранные сегменты, нажать кнопку **OK**.

Резюме

Три приведенных примера демонстрируют наиболее общее применение функций глобального редактирования. При правильном обращении от использования этих мощных функций можно добиться значительной продуктивности. Однако, их мощность и сложность в некоторых случаях способствуют получению непредвиденных результатов, особенно при работе со сложными выделениями.

Если возникают какие-либо сомнения в правильности ожидаемого результата, следует использовать команду снятия выделения со всех объектов (команда **DeSelect All**, горячие клавиши **X, A**), а затем задать более простое выделение. Визуально проконтролировав соответствие критерию выбранных объектов, можно повторно выполнить

глобальное редактирование с использованием атрибута выделения (**Selection**) в качестве критерия.

Для архивации текущего состояния проекта рекомендуется использовать присутствующую в системе Protel 99 SE функцию автосохранения. Особенно это рекомендуется делать в тех случаях, когда пользователь намеревается внести сложные изменения в проект. Наконец, следует помнить, что функции **Undo** и **Redo** при необходимости позволяют восстановить несколько последних состояний.

Перемещение и перетаскивание

При перемещении (**move**) объектов выполняется перестановка их в другое место без учета объектов, находящихся в контакте с перемещаемым объектом. Например, если перемещается некоторый компонент, все соединенные с его контактными площадками проводники перемещены не будут. Если выполняется операция перетаскивания (**drag**) компонента, тогда все проводники останутся подсоединенными к контактными площадкам компонента. Операция перемещения может применяться как к отдельным объектам, так и к группе выделенных объектов, но она применима только к объектам, являющимся частью некоторой цепи.

Выполнение операции перемещения

Для выбора и перемещения какого-нибудь объекта или выделения необходимо:

1. Расположить указатель мыши над перемещаемым объектом.
2. Нажать и удерживать левую кнопку мыши.

В строке статуса появится сообщение о перемещаемом объекте.

3. Переместить объект в новое место.

Выполнение операции перетаскивания

Для перетаскивания объекта, который является частью какой-либо цепи, необходимо:

1. Расположить указатель мыши над перетаскиваемым объектом.
2. Нажать левую кнопку мыши для выделения объекта фокусом.

Если объект является частью цепи, данная цепь будет подсвечена полностью. Если другие объекты не были подсвечены, тогда объект в фокусе не имеет одинакового с ними атрибута цепи и, поэтому, операция перетаскивания не может быть выполнена.

3. Повторно нажать левую кнопку мыши.

Если вы по ошибке щелкнули по метке-манипулятору выделенного фокусом объекта, то при ее перетаскивании будет изменяться размер объекта. Для получения подробной информации о поведении объекта в фокусе обратитесь к пункту *Выделение фокусом* данной главы.

4. Переместить объект в новое место.

Перетаскивание компонента

Для перетаскивания компонента необходимо:

1. Выбрать команду меню **Edit » Move » Drag** (горячие клавиши **M, D**).
2. Щелкнуть левой кнопкой мыши по перемещаемому компоненту и, удерживая ее, перетащить его на новое место.

Поведение проводников, которые соединены с компонентом или проходят под ним, зависит от состояния опции **Component Drag**, находящейся в диалоговом окне **Preferences**, вызываемом командой меню **Tools » Preferences**. Команда **Edit » Move » Drag** также может быть использована для перетаскивания любого графического примитива.

Перемещение группы выделенных объектов

После выделения группа объектов может быть перемещена, как и одиночный объект. С помощью команды меню **Edit » Move** или нажатия на клавишу **M** активизируется меню **Move**. Это меню включает:

Move Selection (перемещение выделения)

Эта команда позволяет выбирать новое местоположение для выделения, которое будет перемещаться как блок (горячие клавиши **M, L**). В процессе работы с этой командой генерируется запрос на выбор точки привязки.

Flip Selection (зеркальное отображение выделения)

Эта команда выполняет зеркальное отображение выделения относительно вертикальной оси. Нажатие клавиши **X** или **Y** при работе с командой **Move Selection** приводит к аналогичному результату.

Rotate Selection (поворот выделения)

Эта команды вызывает диалоговое окно **Rotation Angle**, где задается угол вращения, минимальный шаг которого равен 0.001 градусам. Затем последует запрос на указание точки привязки, которая и будет центром вращения.

Выделение может быть повернуто во время работы с командой **Move Selection**, для чего необходимо нажать клавишу **SPACEBAR**. Угол вращения для таких случаев устанавливается во вкладке **Options** диалогового окна **Preferences**.

При выводе окончательного варианта печатной платы на плоттер в формате Gerber некоторые плоттеры могут не принимать повернутые примитивы. Поэтому, во время генерации файла в формате Gerber повернутые прямоугольные объекты, например прямоугольные контактные площадки компонентов, автоматически закрашиваются с использованием апертур круглой формы.

Перемещение одиночных элементов

Другие остальные команды меню **Edit » Move** работают только с невыделенными объектами. Поскольку эти команды манипулируют непосредственно заданными элементами, становится проще управлять выделением при насыщенном объектами чертеже, когда многие из них перекрывают друг друга.

Перемещение или удаление одного или более элементов может привести к образованию "дыр" на чертеже под этими элементами. Это происходит по той причине, что редактор печатных плат не выполняет постоянную перерисовку содержимого экрана во время перемещения или удаления, так как это значительно ухудшало бы производительность системы. Для обновления экрана применяется кнопка **Redraw**, расположенная на панели инструментов **Tool**, или клавиша **END**.

Вращение объектов против часовой стрелки осуществляется с помощью клавиши **SPACEBAR**, а по часовой стрелке — посредством сочетания клавиш **SHIFT+SPACEBAR**. Угловой шаг вращения устанавливается в диалоговом окне **Preferences**.

Break Track (излом проводника)

Команда **Break Track** преобразует один сегмент проводника, в два сегмента с точкой излома. Для этого необходимо:

1. Выбрать команду **Edit » Move » Break Track** (горячие клавиши **M, B**).

Появится запрос на выбор проводника **Choose a track**.

2. Расположить указатель мыши над сегментом проводника и нажать клавишу **ENTER** или левую кнопку мыши.

Проводник будет отображен в режиме рисования.

3. Переместить указатель мыши в место нового излома.
4. Повторно нажать клавишу **ENTER** или левую кнопку мыши для завершения операции.

Снова появится запрос на выбор проводника **Choose a track**. В процессе перетаскивания можно прервать перемещение с помощью или нажатия клавиши **ESC**. Заметим, что запрос **Select Track** все еще остается на экране.

5. Выбрать другой проводник. В противном случае нажать клавишу **ESC** второй раз или щелкнуть правой кнопкой мыши для выхода из текущего режима.

Polygon Vertices (вершины полигона)

Форма границ полигона может быть изменена посредством перемещения его вершин. Информация о перемещении вершин полигона приведена в пункте *Полигоны* раздела *Объекты редактора печатных плат*.

Непосредственное удаление объектов

Команда меню **Edit » Delete** используется для непосредственного удаления объектов из активного чертежа печатной платы. Такой вид удаления отличается от простых описанных выше команд вырезания (**Cut**) и удаления (**Clear**), при работе с которыми сначала объекты выделяются, а затем уже выбирается само действие (**Cut** или **Clear**). При непосредственном удалении сначала выбирается действие (**Delete**), а затем выполняется щелчок левой кнопкой по нужному объекту.

Если при удалении под указателем мыши оказываются несколько объектов, появляется список, из которого нужно выбрать удаляемый объект.

Операция непосредственного удаления не зависит от текущего выделения. Например, при удалении одиночных проводников те из них, которые являются частью текущего выделения, будут оставлены нетронутыми.

Все выполненные удаления могут быть восстановлены с помощью команды меню **Edit** » **Undo** или комбинации горячих клавиш **ALT+BACKSPACE**. Если производились последовательные удаления нескольких объектов, то они будут восстанавливаться по одному, начиная с элемента, удаленного последним. Команда меню **Edit** » **Redo** является обратной к команде **Undo** и функционирует аналогичным образом.

- ◆ Для удаления выделенного фокусом объекта используется клавиша **DELETE**.
- ◆ Для удаления комплексного выделения объектов используется комбинация клавиш **CTRL+DELETE**.

Измерение расстояний на чертеже печатной платы

Измерение расстояний

Эта функция используется в случаях, когда необходимо выполнить точные измерения расстояния между двумя точками на чертеже печатной платы. После выполнения команды меню **Reports** » **Measure Distance** появится запрос на выбор начальной точки (**Select Measure Start Point**). Далее нужно поместить указатель мыши в нужную начальную точку, щелкнуть левой кнопкой мыши, переместить указатель в конечную точку и снова щелкнуть левой кнопкой мыши. В специальном диалоговом окне появится измеренное расстояние между указанными точками по осям X и Y.

Замечания по использованию функции измерения расстояний:

Если возникают трудности точного позиционирования указателя мыши в требуемых точках, рекомендуется изменить шаг сетки **Snap Grid** (горячая клавиша **G**).

Иногда требуется временно выключить электрическую сетку **Electrical Grid**, например, если указатель мыши перескакивает в центр электрических объектов (горячая клавиша управления электрической сеткой **SHIFT+E**).

Переключение между метрической и дюймовой системами координат выполняется с помощью горячей клавиши **Q**.

Измерение расстояния между графическими примитивами

В процессе разработки печатной платы часто возникает необходимость измерения расстояний между какими-либо двумя графическими примитивами. Функция **Measure Primitives** обеспечивает простой и быстрый путь нахождения наикратчайшего расстояния между границами двух примитивов.

После выбора команды меню **Report** » **Measure Primitives** необходимо щелкнуть по первому, а затем и по второму примитиву. В появившемся диалоговом окне будет

приведен тип, координаты и слой каждого из этих примитивов, а также наикратчайшее расстояние между их ближайшими границами.

Горячие клавиши редактора печатных плат

Вложенные функции редактирования

Редактор чертежей печатных плат имеет мощные функции редактирования, позволяющие выполнять вторую операцию, не прерывая выполнения текущей операции. Такие функции называются вложенными.

Наличие вложенных команд делает процесс проектирования более гибким и интуитивным. Например, при размещении проводника может возникнуть необходимость удаления ранее проложенного сегмента. В этом случае вовсе не обязательно прерывать процесс размещения проводника, достаточно нажать комбинацию горячих клавиш удаления (**E, D**) и удалить требуемый сегмент проводника, после чего щелкнуть правой кнопкой мыши или нажать клавишу **ESC** для завершения процесса удаления. Редактор вернется в режим размещения проводников, и станет возможным добавление новых участков.

Количество дополнительных процессов, которые могут быть запущены без завершения текущего процесса, зависит от потребности каждого из этих незавершенных процессов в программных ресурсах. Для графических процессов глубина вложенности может достигать десяти. При достижении лимита появится соответствующее предупреждение.

◆ Во время выполнения одного процесса, доступ к другому возможен только при использовании клавиатурных комбинаций.

Отмена процесса перерисовки экрана

Каждый раз при изменении масштаба или вида в окне редактора печатных плат, его содержимое будет обновляться для отображения последних изменений. В любой момент процесс обновления экрана можно прервать нажатием клавиши **SPACEBAR**. Этот прием позволяет сэкономить время всякий раз, когда возникает необходимость немедленно изменить масштаб изображения или переместиться в другую точку чертежа без ожидания завершения перерисовки. Для ускорения обновления содержимого экрана рекомендуется также отключать опцию **Transparent Layers** на вкладке **Display** диалогового окна **Preferences**, а также включать режим **Draft** для всех графических примитивов на вкладке **Show/Hide** этого окна.

◆ Для перерисовки только текущего слоя используется комбинация горячих клавиш **ALT+END**.

Горячие клавиши для мыши

В системе Protel горячие клавиши могут быть как для работы с клавиатурой, так и для работы с мышью. Например, нажатие на клавиш **P, P** позволяет разместить контактную площадку вместо того, чтобы обращаться к меню Place для выбора пункта меню **Pad**, что приведет к тому же результату. Использование левой кнопки мыши вместо клавиши **ENTER** и правой кнопки мыши вместо клавиши **ESC** позволяет выполнять многие операции без использования клавиатуры. В разного рода диалоговых окнах наоборот удобно использовать клавиши **ENTER** и **ESC** вместо нажатий мышью на кнопки **OK** и **CANCEL**. Для некоторых операций, где нежелательно перемещение мыши по листу схемы, работа с клавиатурой выглядит эффективнее. Например, включение **Snap Grid** при размещении текстовой строки (**V, G**) или изменение масштаба обзора при перемещении группы выделенных объектов.

Кроме того, имеется возможность создания пользовательских горячих клавиш, процесс определения которых описан в главе *Работа в среде проектирования Design Explorer*.

- ◆ Двойной щелчок левой кнопкой мыши на любом объекте чертежа приведет к открытию соответствующего этому объекту диалогового окна **Change**, где можно отредактировать его атрибуты.
- ◆ Для перемещения объекта, нужно просто щелкнуть на нем левой кнопкой мыши и, удерживая ее, переместить указатель в новое место.
- ◆ Для удаления графического примитива с чертежа необходимо щелкнуть на нем левой кнопкой мыши, то есть выбрать, а затем нажать клавишу **DELETE**.

Инструмент Slider Hand (рука)

Редактор печатных плат содержит мощное средство перемещения по чертежу, называемое **Slider Hand** (рука). Щелчок правой кнопкой мыши и удержание ее приводит к изменению вида указателя мыши, после чего изображение в окне можно просто передвинуть.

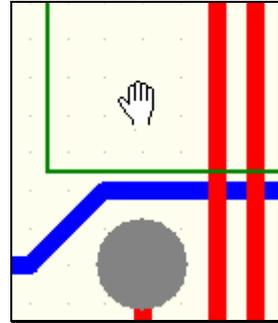


Рис. 6.18. Для быстрого перемещения по чертежу используется инструмент Slider hand

Горячие клавиши для клавиатуры

Существует два способа создания горячих клавиш для клавиатуры. Первым является создание в специальном редакторе **Keyboard Shortcut Editor**, вызываемого с помощью команды меню **Client Menu » Customize**. Здесь создаются горячие клавиши, которые непосредственно запускают тот или иной процесс. Например, при нажатии клавиш **CTRL+G** появляется диалоговое окно **Snap Grid**, позволяющее изменить шаг сетки размещения.

Какой-либо процесс может быть запущен с помощью так называемых горячих клавиш меню. Ряд команд меню содержит подменю, которые открываются при их выборе, другие команды подменю не имеют и запускают соответствующую процедуру. Например, нажатие на клавишу **P** вызывает меню **Place**, а последующее нажатие на **V** позволяет разместить на листе переходное отверстие. Нажатие на клавишу **T** вызывает меню **Tools**, а последующее нажатие на **R** вызовет всплывающее меню **Auto Route**.

◆ Горячие клавиши для клавиатуры имеют более высокий приоритет, чем горячие клавиши меню, поэтому при совпадении комбинаций будут срабатывать именно клавиатурные.

Горячие клавиши вызова команд меню:

A	Auto Route (автотрассировка)
B	View » Toolbars (панель инструментов)
D	Design (проект)
E	Edit (редактирование)
F	File (файл)
G	Snap Grid (сетка размещения)
H	Help (помощь)
I	Interactive placement (интерактивное размещение)
J	Edit » Jump (переход)
M	Edit » Move (перемещение)

N	Netlist (список соединений)
O	Options (настройка)
P	Place (размещение)
R	Reports (отчет)
S	Edit » Select (выделение)
T	Tools (инструменты)
U	Tools » Unroute (не трассировать)
V	View (обзор)
W	Window (окно)
X	Edit » DeSelect (отмена выделения)
Z	Zoom (масштаб)

Горячие клавиши:

L	Вкладка Layers диалогового окна Document Option
Q	Переключение единиц измерения
CTRL+D	Вкладка Display диалогового окна Preferences
CTRL+G	Диалоговое окно Snap Grid
CTRL+H	Команда Edit » Select» Connected Copper
CTRL+L	Вкладка Layers диалогового окна Document Option
CTRL+M	Измерить расстояние
CTRL+O	Вкладка Options диалогового окна Preferences
PGUP	View » Zoom Int (увеличить масштаб)
PGDN	View » Zoom Out (уменьшить масштаб)
CTRL+PGUP/PGDN	Максимальный/минимальный масштаб
SHIFT+PGUP/PGDN	Масштабирование с шагом 0.1 от стандартного
HOME	Команда View » Pan
END	Команда View » Refresh (обновить чертеж)
CTRL+HOME	Переход к абсолютному началу координат
CTRL+END	Переход к текущему началу координат
CTRL+INS	Команда Edit » Copy (копировать в буфер обмена)
CTRL+DEL	Команда Edit » Clear (удалить)
SHIFT+INS	Команда Edit » Paste (вставить из буфера обмена)
SHIFT+DEL	Команда Edit » Cut (вырезать в буфер обмена)
ALT+BACKSPACE	Undo (отменить)
CTRL+BACKSPACE	Redo (вернуть)
SHIFT+E	Включить/выключить сетку
SHIFT+S	Включить/выключить однослойный режим
SHIFT+R	Циклическое переключение режимов трассировки
SHIFT+F4	Расположить все открытые окна каскадом
SHIFT+F5	Расположить все открытые окна мозаикой
*	Последовательный перебор активных сигнальных слоев
+ или -	Последовательный перебор активных слоев

F1	Вызов контекстной справки
UP, DOWN	Перемещение курсора вверх или вниз (на 1 шаг)
shIFT+UP, DOWN	Перемещение курсора вверх или вниз (на 10 шагов)
LEFT, RIGHT	Перемещение курсора влево или вправо (на 1 шаг)
SHIFT+LEFT, RIGHT	Перемещение курсора влево или вправо (на 10 шагов)

Специальные горячие клавиши, функционирование которых зависит от текущего режима работы

TAB	Открытие диалогового окна редактирования атрибутов размещаемого в данный момент объекта.
SPACEBAR	Переключение режимов размещения дорожек; вращение элемента против часовой стрелки при перемещении (шаг поворота задается в диалоговом окне Preference); прерывание процесса обновления экрана; изменение увеличения мини-браузера.
SHIFT	Ускорение перемещения по чертежу (скорость задается в диалоговом окне Preference).
SHIFT+SPACEBAR	Переключение режимов размещения дорожек; вращение элемента против часовой стрелки при перемещении.
N	Оптимизация списка соединений после перемещения компонента.
CTRL	Временное отключение электрической сетки Electrical grid при редактировании.
CTRL+SPACEBAR	Циклический перебор всех линий связи указанной контактной площадки

Размещение компонентов

Часто возникает ситуация, когда компонент, который необходимо отредактировать или переместить, в настоящий момент не виден на экране. Например, нужно разместить какой-либо уже существующий компонент в место на чертеже, где вы сейчас находитесь, но изменять масштаб или вид для его поиска не желательно. В этом случае можно поступить следующим образом: выбрать пункт меню **Edit » Move » Move Component** (горячие клавиши **M, C**); произвести щелчок левой кнопкой мыши в свободном от компонентов месте чертежа, после чего появится диалоговое окно **Component Designator**, где нужно ввести позиционное обозначение и нажать кнопку **OK**. Если компонент окажется вне экрана, нужно прокрутить изображение таким образом, чтобы переместить курсор мыши и компонент в поле зрения.

Во всплывающем диалоговом окне **Component Designator** можно также поставить символ-заменитель **?** и нажать кнопку **OK**. Это приведет к появлению диалогового окна **Component Placed** позволяющего выбрать любой компонент, находящийся на чертеже платы.

Операции Undo и Redo

Редактор печатных плат имеет многоуровневый механизм отмены и возврата к состоянию, предшествующему операции отмены. Каждое действие сохраняется в специальной области памяти по стековому принципу. При выполнении команды **Undo** из стека выбирается последняя операция.

Редактор печатных плат оптимизирован таким образом, чтобы операции **Undo** и **Redo** обеспечивали минимальный расход памяти, однако, существует возможность очистить стек этих команд, временно присвоив в диалоговом окне **Preferences** параметру **Stack Size** (размер стека) значение равное нулю.

Объекты редактора печатных плат

Среда проектирования Protel 99 SE и при разработке печатной платы, и при работе в редакторе библиотек топологических посадочных мест состоит из двух основных компонентов: объектов, размещаемых на чертеже и являющихся строительным материалом любого проекта, и процедур, выполняемых системой или пользователем для создания, изменения, сохранения объектов, а также для составления отчетов.

В редакторе чертежей печатных плат существует два типа объектов: Примитивы и Составные объекты. Примитивы — это базовые топологические элементы, включающие как электрические объекты (проводники, контактные площадки, переходные отверстия и области металлизации), так и графические объекты (текст, линии и дуги). Составные объекты, например, топологические посадочные места компонентов и размеры, строятся из групп электрических и графических объектов (примитивов).

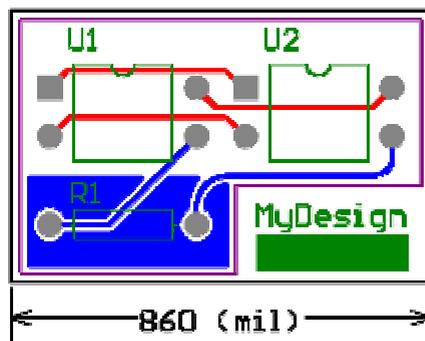


Рис. 6.19. Чертеж платы представляет собой набор графических примитивов.

Графические примитивы

Линии

Линии являются основными графическими элементами на чертежах печатных плат. Они могут размещаться на любом слое и иметь ширину от 0.001 до 10000 мил (от 0.03 мкм до 254 мм). Если они располагаются на сигнальном слое, то они играют роль проводников, то есть являются электрическими объектами. На механических слоях с помощью них можно задавать границы платы и областей, где трассировка проводников запрещена.

◆ Для размещения линии на механическом слое вместо команды **Place » Interactive Routing** используется команда **Place » Line**. Внешне ничего не изменится, однако отличие будет в том, что параметры примитивов **Line** задаются на вкладке **Default Primitives** диалогового окна **Preferences**, а проводников (**Track**) — определяются правилами проектирования.

На слое шелкографии линиями прорисовываются контуры компонентов. Линии являются основными графическими элементами на чертежах печатных плат. Они могут размещаться на любом слое и иметь ширину от 0.001 до 10000 мил (от 0.03 мкм до 254 мм). Если они располагаются на сигнальном слое, то они играют роль проводников, то есть являются электрическими объектами. На механических слоях с помощью них можно задавать границы платы и областей, где трассировка проводников запрещена. На слое шелкографии линиями прорисовываются контуры компонентов.

Размещение проводников

Так как основной интерес при разработке плат представляет размещение проводников, рассмотрим процесс размещения линий как проводников. Каждый проводник должен располагаться на определенном слое, для чего надо выбрать нужный слой, щелкнув левой кнопкой мыши на вкладке в нижней части среды проектирования. Для размещения сегмента проводника в текущем слое необходимо:

1. Выполнить команду меню **Place » Interactive Routing**, нажать горячие клавиши **P, T** или щелкнуть по кнопке **Track**, расположенной на панели **Placement Tools**.

При этом в строке состояния появится запрос **Choose start location** (выберите начальную точку).

2. Щелкнуть левой кнопкой мыши или нажать клавишу **ENTER** для задания начальной точки дорожки.

◆ В режиме размещения проводников пользователь может:

- последовательно переключать сигнальные слои путем нажатия клавиши * на цифровой клавиатуре.
- последовательно переключать все существующие слои с помощью клавиш + и - на цифровой клавиатуре.

При этом в строке состояния будет отображаться имя цепи соответствующей данному проводнику, длина текущего сегмента и общая длина проводника (в скобках).

3. Переместить указатель мыши в любом направлении. При выполнении этих операций следует обратить внимание на способ отображения двух сегментов проводника: один сегмент сплошной, а другой в виде контура. Сплошной сегмент проводника является размещаемым в данный момент, а ооконтуренный — предсказываемый.

мым. Режим трассировки с предсказанием подробно описан в разделе *Ручная трассировка печатной платы*.

По умолчанию инициализируется стиль размещения изломов проводников **45 Degree End**. В этом стиле первый размещаемый сегмент будет находиться в вертикальном или горизонтальном положении, а следующий сегмент будет располагаться под углом 45° к нему. При нажатии клавиши **SPACEBAR** произойдет переключение в режим **45 Degree Start**, в котором текущий сегмент размещается по диагонали, а предсказываемый — вертикально или горизонтально. Повторное нажатие клавиши **SPACEBAR** приводит к возврату в режим **45 Degree End**.

4. Для размещения первого сегмента проводника необходимо переместить курсор в то место, где будет находиться конец этого сегмента, и щелкнуть левой кнопкой мыши (или нажать клавишу **ENTER**).

5. Дальнейшее перемещение курсора приведет к прокладыванию нового сегмента проводника, от размещенного ранее. Режимы прокладывания проводников переключаются нажатием клавиши **SPACEBAR**.

Каждый раз при задании нового сегмента проводника необходимо производить щелчок левой кнопкой мыши.

6. Щелчок правой кнопкой мыши приведет к завершению последовательного соединения участков проводников.

Обратите внимание, что в строке состояния все еще присутствует сообщение **Choose start location** (выберите начальную точку). Это означает, что пользователь завершил прокладывание одной последовательности сегментов проводника и готов к прокладыванию новой в другом месте платы без повторного выполнения команды **Place » Interactive Routing**.

7. Повторное нажатие клавиши **ESC** или щелчок правой кнопкой мыши приведет к выходу из режима размещения проводников.

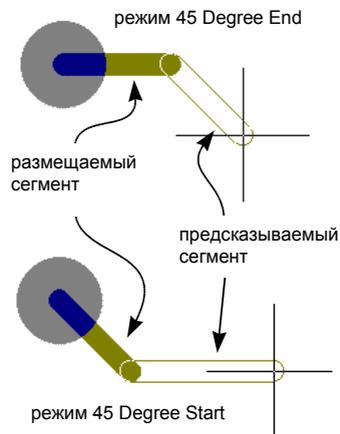


Рис. 6.20. Прорисовка проводников с предсказанием

◆ Неправильно проложенный сегмент проводника удаляется нажатием клавиши **BACKSPACE**.

◆ Для редактирования параметров проводника во время размещения, необходимо нажать клавишу **TAB**, что приведёт к появлению диалогового окна **Track**. В случае, если линия прокладывается на сигнальном слое, то есть прокладывается проводник, имеется возможность изменения только ее ширины в пределах между минимальным и максимальным значением, заданными в правиле проектирования **Width Constraint** для данной цепи.

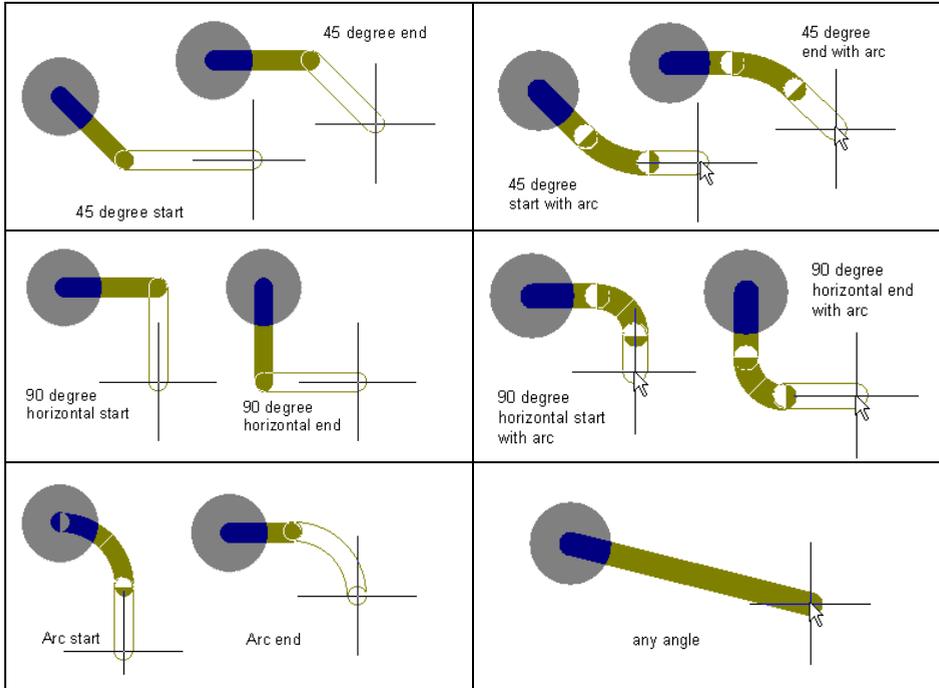


Рис. 6.21. Шесть режимов размещения проводников

Режимы размещения проводников

Редактор печатных плат имеет шесть режимов размещения проводников, которые различаются способом реализации изгибов (рис. 6.21):

45 Degree (Start and End)

Проводники могут располагаться горизонтально и вертикально, угол изгиба не превышает 45° .

45 Degree with Arc (Start and End)

Проводники могут располагаться горизонтально и вертикально, угол изгиба не превышает 45° , сам изгиб выполняется закругленным.

90 Degree (Start and End)

Проводники могут располагаться горизонтально и вертикально, угол изгиба может составлять 90° .

90 Degree with Arc (Start and End)

Проводники могут располагаться горизонтально и вертикально, угол изгиба может составлять 90° , сам изгиб выполняется закругленным.

Any Angle

Сегменты проводников могут размещаться под произвольным углом.

Arc Start and Arc End

Проводники могут располагаться горизонтально или вертикально с дугами под углом 90° на концах.

◆ Переключение режимов **Start** и **End** производится нажатием клавиши **SPACEBAR**. Одновременное нажатие **SHIFT+SPACEBAR** последовательно переключает режимы размещения проводников. Нажатие клавиш “.” и “,” соответственно увеличивает и уменьшает размер дуги в режимах использующих дуговые сегменты.

Размещение проводников для трассировки соединений

Размещение проводника, начинающегося с объекта, для которого назначено имя цепи, (например, контактной площадки) называется трассировкой этой цепи. Размещаемый проводник получит имя цепи, назначенное контактной площадке. Все правила проектирования, назначенные данной цепи, будут также соблюдаться. При размещении проводников для трассировки соединений используются некоторые функции, упрощающие эту процедуру. Более подробная информация по этому вопросу приведена в разделе *Ручная трассировка печатной платы*.

Редактирование проводников

Проводники могут быть изменены как по отдельности, так и целиком все вместе. Для редактирования существующих дорожек нужно выбрать пункт меню **Edit » Change**. Щелчок левой кнопкой мыши по проводнику вызывает диалоговое окно **Change Track**, в котором производится изменение его атрибутов.

Параметры проводников, задаваемые по умолчанию

Во время размещения проводника его параметры можно изменить в диалоговом окне **Track**, вызываемом нажатием клавиши **TAB**. Устанавливаемые по умолчанию параметры проводников задаются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Примечание: в случае, если проводник является участком цепи, имеется возможность изменения его ширины только в пределах между минимальным и максимальным значением, заданными для данной цепи правилом проектирования **Width Constraint**.

Контактные площадки

Контактные площадки могут быть многослойными или однослойными, причем в последнем случае они могут размещаться на любом. Например, компоненты для поверхностного монтажа и краевые разъемы имеют однослойные контактные площадки на верхнем или нижнем слоях. Контактные площадки могут иметь форму круга,

прямоугольника, закругленного прямоугольника или восьмиугольника с размерами от 1 до 10000 мил (от 0.0254 до 254 мм). Размер отверстия может находиться в диапазоне от 0 (для компонентов поверхностного монтажа) до 1000 мил (от 0 до 25.4 мм). Длина обозначения контактных площадок не должна превышать 4 символов.

В случае многослойной контактной площадки, форма и размер для верхнего, нижнего и промежуточных слоев могут быть назначены независимо при определении стека контактной площадки. Контактные площадки могут использоваться как в составе компонентов, так и по отдельности.

Размещение контактных площадок

Отдельные контактные площадки (то есть не входящие в состав какого-либо компонента) могут быть расположены в любом месте чертежа печатной платы. Контактные площадки со сквозными отверстиями, как и переходные отверстия, являются многослойными объектами, которые присутствуют на всех сигнальных слоях платы и размещаются без указания конкретного слоя. Однослойные контактные площадки могут быть размещены на любом слое платы.

Для размещения контактной площадки необходимо выполнить команду меню **Place » Pad** (горячие клавиши **P, P**) или нажать кнопку **Pad** на панели инструментов **PlacementTool**.

Обозначение контактной площадки

Контактные площадки могут иметь обозначение, обычно представляющим собой номер вывода компонента, длина которого не должна превышать 4 символов. Использование пробелов не допускается.

Если первая контактная площадка имеет в обозначении число, то при размещении следующих площадок численная часть обозначения будет автоматически увеличиваться на единицу. Принудительное присвоение обозначения выполняется во время размещения контактной площадки после нажатия клавиши **TAB**.

При необходимости увеличения численной части обозначения контактной площадки на величину, отличную от 1, следует использовать функцию **Paste Array**. Для этого контактную площадку надо скопировать в буфер обмена, а при вставке массива в диалоговом окне **Paste Array** установить нужное значение в поле **Text Increment**. Допускаются следующие последовательности обозначения контактных площадок:

Цифровая (1, 3, 5);

Буквенная (A, B, C);

Комбинация букв и цифр (A1, A2... или 1A, 1B... или A1, B1... или 1A, 2A... и т. д.)

При увеличении численной части обозначения в поле **Text Increment** просто вводится нужное число. При увеличении буквенного обозначения в это поле надо ввести букву, обозначающую количество пропускаемых букв. Например, если начальная контактная площадка имеет обозначение **1A**, а в поле **Text Increment** установлено **C** (третья буква алфавита), то последовательность обозначений будет следующей: **1A**, **1D** (третья буква после A), **1G** (третья буква после D) и так далее.

Редактирование контактных площадок

И отдельные контактные площадки, и контактные площадки, входящие в состав компонентов, могут быть изменены как индивидуально, так и глобально. Для редактирования контактной площадки нужно выбрать пункт меню **Edit » Change**, после чего в строке состояния появится подсказка **Change Any Object** (изменение произвольного объекта). Щелчок левой кнопкой мыши по контактной площадке вызовет диалоговое окно, в котором осуществляется редактирование параметров контактной площадки.

Параметры контактных площадок, задаваемые по умолчанию

Параметры размещаемой в данный момент контактной площадки могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » Pad**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Переходные отверстия

Переходные отверстия используются в случае необходимости соединения проводников, расположенных на разных слоях. Переходные отверстия подобны круглым контактным площадкам со сквозным отверстием, в которое при производстве платы наносится слой металлизации.

Переходные отверстия могут быть многослойными, глухими или скрытыми и иметь диаметр от 2 до 10000 мил (от 0.0508 до 254 мм). Переходные отверстия могут размещаться принудительно вручную с помощью кнопки **Via**, расположенной на панели инструментов **PlacementTool**, или команды меню **Place » Via**, а также автоматически при интерактивной трассировке проводников или в процессе автотрассировки. Размер отверстия может находиться в диапазоне от 0 до 1000 мил (от 0 до 25.4 мм).

Типы переходных отверстий

Переходные отверстия могут быть многослойными, глухими или скрытыми. Многослойные переходные отверстия обеспечивают сквозное соединение всех внутренних сигнальных слоев. Глухое переходное отверстие соединяет один из наружных слоев платы (верхний или нижний) с внутренним слоем, а скрытые переходные отверстия соединяют между собой два внутренних слоя.

Глухие и скрытые переходные отверстия

При использовании глухих и скрытых переходных отверстий разработчик должен ясно представлять технологию изготовления плат, применяемую на предполагаемом предприятии-изготовителе. Большинство производителей обеспечивают выполнение глухих и скрытых переходных отверстий между так называемыми парными слоями. При такой технологии многослойная плата выполняется в виде набора спрессованных между собой тонких двухсторонних плат. Это позволяет с помощью глухих и скрытых переходных отверстий реализовывать соединения между поверхностями этих тонких двухсторонних плат, в результате чего образуются парные слои. Конфигурирование слоев производится в диалоговом окне **Layer Stack Manager**. Подробности по этому вопросу приведены в разделе *Описание печатной платы*.

Как только пользователь определит стек слоев, ему необходимо назначить пары слоев сверления. Пары слоев сверления определяются в диалоговом окне **Drill Pair Manager**, вызываемом нажатием кнопки **Drill Pair** в окне **Layer Stack Manager**. После определения таких слоев, редактор печатных плат будет автоматически вставлять проходные отверстия нужного типа между слоями трассировки.

Параметры переходных отверстий, задаваемые по умолчанию

Параметры размещаемого в данный момент переходного отверстия могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » Via**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Размещение переходных отверстий

Переходные отверстия могут быть размещены на плате программой автотрассировки, а также автоматически средствами интерактивной трассировки и принудительно вручную.

Переходные отверстия, размещаемые автотрассировщиком

Автотрассировщик выполняет размещение переходных отверстий согласно установленным для платы правилам проектирования **Routing Via Style** (команда меню **Design » Rules**). Подробное описание этого правила проектирования приведено в разделе *Определение требований к проекту печатной платы*.

Функция автоматического размещения переходных отверстий

Нажатие клавиши * для перехода сигнальный слой во время процесса размещения проводника, приводит к автоматическому добавлению на плату переходного отверстия. Параметры этого переходного отверстия будут удовлетворять заданному стилю, установленному правилом **Routing Via Style**, и паре слоев сверления. Для изменения параметров отверстий используется клавиша **TAB**.

Размещение переходных отверстий в ручном режиме

Для размещения переходного отверстия вручную необходимо выполнить команду меню **Place » Via** и мышью указать его местоположение.

Редактирование переходных отверстий

Для редактирования переходного отверстия нужно выполнить команду меню **Edit » Change**. Щелчок левой кнопкой мыши на нужном переходном отверстии приводит к появлению диалогового окна **Change Via**, в котором осуществляется редактирование его параметров.

Соединение переходных отверстий с внутренними слоями питания и заземления

Стиль соединения переходного отверстия с внутренними слоями питания и заземления задается аналогично контактным площадкам правилом проектирования **Power Plane Connect Style**, определенным для данной цепи.

Если пользователь не хочет, чтобы данное переходное отверстие было соединено с соответствующим внутренним слоем, ему необходимо определить новое правило проектирования **Power Plane Connect Style** только для данного объекта и выбрать стиль **No Connect**. Подробности по использованию правил проектирования приведены в разделе *Определение правил проектирования*.

Прямоугольные области заливки

Прямоугольные области заливки могут быть размещены на любом слое платы. При их размещении в сигнальном слое они становятся сплошными проводящими областями металлизации, которые могут использоваться для экранировки или в качестве проводника обеспечивающего протекание больших токов. Для покрытия участков неправильной формы можно использовать несколько наложенных друг на друга прямоугольных областей заливки. При их объединении с проводниками они будут распознаны программой проверки правил проектирования как элементы соответствующей цепи.

Области заливки могут быть размещены на непроводящих слоях. Например, размещение области заливки в слое **Keep Out** может быть использовано для обозначения участка недоступного как для автотрассировки, так и для автоматического размещения компонентов. Области заливки могут использоваться для создания пустот в слоях питания и заземления, а также, **Solder Mask** или **Paste Mask**.

◆ В случае если первый угол области помещен на объекте, имеющем имя цепи, то область заливки унаследует это имя.

Чтобы создать область заливки, необходимо:

1. Выполнить команду меню **Place » Fill**, после чего в строке состояния появится подсказка **Select First Corner** (выберите первый угол).
2. Расположите указатель мыши в нужном месте чертежа и щелкните левой кнопкой мыши.
3. Переместите курсор в противоположный угол.
4. Щелкните левой кнопкой мыши для задания второго угла.
5. Для выхода из режима размещения необходимо щелкнуть правой кнопкой мыши.

Редактирование областей заливки

Для редактирования области заливки нужно выбрать команду меню **Edit » Change**. После щелчка левой кнопкой мыши на области заливки появится диалоговое окно **Fill**, в котором осуществляется редактирование ее параметров.

Параметры областей заливки, задаваемые по умолчанию

Параметры размещаемой в данный момент области заливки могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » Fill**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Дуги

С помощью дуг на платах прорисовываются закругленные участки проводников. Дуги могут размещаться на любых слоях и иметь радиус от 0.001 до 16000 мил (от 0.0254 мкм до 406.4 мм) и ширину от 0.001 до 10000 мил (от 0.0254 мкм до 254 мм). Угловое разрешение составляет 0.001° . Для размещения дуги на чертеже необходимо выполнить команду меню **Place » Arc** или нажать кнопку **Arc** панели инструментов **PlacementTools**. Дуги могут быть прорисованы как продолжение проводника, выполняемого с помощью команды **Place Track**, а также как граница залитого полигона сложной формы.

В редакторе чертежей печатных дуги находят самое разнообразное применение. Например, они могут быть использованы для прорисовки контура компонента на слое шелкографии или контура платы и штампованных отверстий на механических слоях. Дуги могут быть как незамкнутые, так и замкнутые, которые используются для создания окружностей.

◆ При размещении проводников имеется возможность быстрого вызова прорисовки дуг. Для этого требуется, чтобы был выбран такой режим размещения проводников, который допускает использование дуг. Переключение режимов размещения производится нажатием комбинации клавиш **SHIFT+SPACEBAR**. После выбора требуемого режима размещения дуги, переключение между режимами Start и End производится с помощью клавиши пробел.

Размещение дуг (первая точка — центр)

Размещение дуг на текущем слое, с первой точкой в центре дуги, выполняется следующим образом:

1. Выполните команду **Place » Arc (Center)** (горячие клавиши **P, A**) или нажмите кнопку **Arc** на панели инструментов **PlacemantTool**. в строке состояния появится подсказка **Select Arc Center** (выберите центр дуги). Во время выполнения этой операции нажатием клавиши * можно изменить сигнальный слой, а последовательно нажатием клавиш + или – перейти на любой из активных слоев платы.
- ◆ Смена направления дуги производится нажатием клавиши **SPACEBAR** перед вводом последней точки.
2. Переместите указатель мыши в точку предполагаемого центра дуги и выполните щелчок левой кнопкой мыши. Теперь дуга будет прорисовываться по мере перемещения указателя мыши.
 3. Далее перемещением указателя мыши задается радиус дуги.
 4. После задания радиуса щелчком левой кнопки мыши задается начальная точка дуги.
 5. Затем аналогичным образом выбирается конечная точка дуги. Для прорисовки окружности необходимо определить начало и конец дуги в одной точке.

6. Выход из режима рисования дуг осуществляется щелчком правой кнопкой мыши или нажатием клавиши **ESC**.

Размещение дуг (первая точка — начало дуги)

Размещение дуг на текущем слое, с первой точкой в начале дуги, выполняется следующим образом:

1. Выполните команду меню **Place » Arc (Edge)** (горячие клавиши **P, E**).
2. Переместите указатель мыши в точку начала дуги и выполните щелчок левой кнопкой мыши.
3. Переместите указатель мыши в конец дуги и выполните щелчок левой кнопкой мыши.
4. Выход из режима рисования дуг осуществляется щелчком правой кнопкой мыши или нажатием клавиши **ESC**.

Размещение дуг, имеющих произвольный угол (первая точка — начало дуги)

Размещение дуг, имеющих произвольный угол, на текущем слое, с первой точкой в начале дуги, выполняется следующим образом:

1. Выполните команду меню **Place » Arc (Any Angle)** (горячие клавиши **P, N**).
2. Переместите указатель мыши в точку начала дуги и выполните щелчок левой кнопкой мыши.
3. Указателем мыши задайте радиус и центр дуги.
4. Переместите указатель мыши в конец дуги и выполните щелчок левой кнопкой мыши.
5. Выход из режима рисования дуг осуществляется щелчком правой кнопкой мыши или нажатием клавиши **ESC**.

Размещение окружностей

Для размещения окружности (замкнутой дуги):

1. Выполните команду меню **Place » Full Circle** (горячие клавиши **P, U**).
2. С помощью указателя мыши задайте центр окружности и выполните щелчок левой кнопкой мыши.
3. Указателем мыши задайте радиус окружности и выполните щелчок левой кнопкой мыши.
4. Выход из режима рисования окружностей осуществляется щелчком правой кнопкой мыши или нажатием клавиши **ESC**.

Редактирование дуг

Для редактирования дуги нужно выбрать команду меню **Edit » Change**. После щелчка левой кнопкой мыши на нужной дуге появится диалоговое окно **Arc**, в котором осуществляется редактирование ее параметров.

Параметры дуг, задаваемые по умолчанию

Параметры размещаемой в данный момент дуги могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » Arc**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Текстовые строки

Редактор топологий печатных плат допускает размещение на любом из слоев текстовых строк длиной не более 254 символа (включая пробелы) и высотой от 0.010 до 10000 мил (от 0.254 мкм до 254 мм). Для размещения текстовой строки на чертеже необходимо выполнить команду меню **Place » String** или нажать кнопку **Text String** на панели инструментов **PlacementTools**.

Текст на чертежах печатных плат формируется с помощью трех специальных шрифтов. По умолчанию используется простой векторный шрифт **Default**, который может быть легко прорисован перьевыми плоттерами и векторными фотоплоттерами. Использование двух других более сложных шрифтов **Sans Serif** и **Serif** приводит к замедлению вывода файлов в векторном формате (например, в формате Gerber). Эти шрифты встроены в программное обеспечение и не могут быть изменены. Все шрифты имеют расширенный набор ASCII символов, что обеспечивает поддержку английского и других европейских языков.

Все текстовые строки (например, обозначения и описания компонентов или просто надписи) имеют одинаковый набор параметров, поэтому их редактирование и перемещение выполняется одним и тем же способом. Текстовые строки, не связанные с какими-либо объектами могут быть размещены на любом из слоев. Текстовые строки, связанные с компонентами, при размещении их на плате автоматически помещаются в верхний или нижний слой шелкографии (**Overlay**), но затем могут быть перемещены на любой другой слой платы.

Простые текстовые строки могут быть перемещены и отредактированы подобно любым другим графическим примитивам. Текстовые строки, связанные с компонентами, могут быть перемещены независимо от самих компонентов с помощью команды **Edit » Move**. При перемещении самого компонента его текстовые строки будут перемещаться с сохранением своего относительного расположения.

Редактор чертежей печатных плат имеет возможность использования специальных строк, содержимое которых интерпретируется при выводе чертежа на печать. Подробности по этому вопросу будут приведены несколько позже.

Параметры текстовых строк, задаваемые по умолчанию

Параметры размещаемой в данный момент текстовой строки могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » String**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Размещение текстовых строк

Размещение не связанной с каким-либо компонентом текстовой строки на текущем слое выполняется следующим образом:

1. Выполните команду меню **Place » String**, после чего новая строка окажется захваченной указателем мыши.
2. Нажмите клавишу **TAB** для вызова диалогового окна **Change String**.
3. Введите текст надписи в поле **Text** или выберите одну из специальных строк, перечисленных в выпадающем списке.
4. Установите требуемые высоту букв, толщину линий и шрифт.
5. Нажмите кнопку **OK**. Выберите местоположение строки и выполните щелчок левой кнопкой мыши.
6. Для выхода из режима размещения текстовых строк необходимо щелкнуть правой кнопкой мыши или нажать клавишу **ESC**.

◆ Зеркальное отображение текстовых строк производится нажатием клавиш **X** и **Y**, а поворот — нажатием клавиши **SPACEBAR**.

Редактирование текстовых строк

Не связанные с компонентами текстовые строки могут быть отредактированы индивидуально или глобально. Двойной щелчок левой кнопкой мыши на строке вызывает диалогового окна **String**, в котором можно отредактировать ее параметры.

Специальные строки

Специальные строки позволяют вводить на чертеж служебную информацию. Например, строка **.PRINT_DATE** во время вывода на печать будет заменена текущей датой. Ниже приведен список разрешенных специальных строк:

.PRINT_DATE
.PRINT_TIME
.PRINT_SCALE
.LAYER_NAME
.PCB_FILE_NAME
.PCB_FILE_NAME_NO_PATH
.PLOT_FILE_NAME
.ARC_COUNT
.COMPONENT_COUNT
.FILL_COUNT
.HOLE_COUNT
.NET_COUNT
.PAD_COUNT
.STRING_COUNT

.TRACK_COUNT
 .VIA_COUNT
 .DESIGNATOR
 .COMMENT
 .LEGEND
 .NET_NAMES_ON_LAYER

Текстовые строки **.DESIGNATOR** и **.COMMENT** можно добавлять к компоненту при создании его в редакторе библиотек топологических посадочных мест. В этом случае надо заранее задать их положение на чертеже и слой, так как это нельзя изменить в редакторе чертежей. Стандартные текстовые строки в этом случае можно скрыть.

Размещенная в слое **Drill Drawing** специальная строка **.LEGEND** при выводе на печать будет заменена на таблицу сверления.

Для отображения содержимого (а не имени) специальных строк на экране необходимо включить опцию **Convert Special String** на вкладке **Display** диалогового окна **Preferences**. Отметим, что не все специальные строки могут быть отображены на экране во время работы с проектом.

Групповые объекты

Групповыми объектами называются наборы примитивов, обрабатываемые как единый объект. Групповые объекты могут создаваться пользователем, например компоненты или полигоны, или системой, например координатные метки и размеры. Их можно размещать, выделять, копировать, изменять, перемещать и удалять.

◆ Для разбиения полигона на отдельные графические примитивы используется команда **Tools » Convert » Explode Polygon to Free Primitives**.

Полигоны

Полигонами называются области металлизации неправильной формы (как правило, достаточно сложной формы), которые могут состоять из одной или нескольких частей, соединенных с заданной цепью. Размещение полигонов производится с помощью команда меню **Place » Polygon Plane**.

Несмотря на то, что полигоны состоят из набора линий и дуг, все манипуляции с ними производятся как с единым объектом. После размещения, границы полигона могут быть изменены, а сами полигоны могут быть обновлены для обтекания новых препятствий. Кроме этого, изменен может быть любой из их параметров, например, ширина линий штриховки

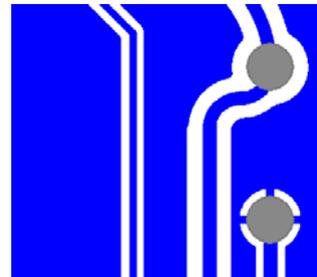


Рис. 6.22. Пример полигона.

и шаг их расположения, благодаря чему можно реализовать или заштрихованный, или полностью залитый полигон.

Обтекание существующих объектов (проводников, контактных площадок, переходных отверстий, областей заливки, надписей) производится в строгом соответствии с назначенными правилами проектирования, определяющими зазоры между проводящими частями различных цепей (рис 6.22). При назначении полигону некоторой цепи, все контактные площадки, находящиеся внутри него, будут автоматически соединены с ним.

Полигоны могут быть размещены на любом слое платы. Если полигон размещается на не сигнальном слое, он не будет обтекать существующие объекты, так как в этом случае ни они, ни сам полигон не будут соединены с какой либо цепью.

Размещение полигонов

Для размещения полигона на текущем слое необходимо выполнить следующие действия:

1. Выполните команду меню **Place » Polygon Plane** (горячие клавиши **P, G**), после чего появится диалоговое окно **Place Polygon Plane**, которое позволяет установить нужные параметры полигоны. Описание параметров будет приведено позднее.
2. Переместите указатель мыши в начальную точку полигона и выполните щелчок левой кнопкой мыши.

3. Продолжайте последовательно вводить точки излома границы полигона, пока он не будет задан полностью. Для изменения режима размещения границы полигона используется клавиша **SPACEBAR**.

Как только граница будет замкнута, будет произведена заливка полигона. Если в настоящий момент граница полигона не замкнута, то по нажатию клавиши **ESC** или щелчку правой кнопкой мыши произойдет автоматическое замыкание полигона линией, соединяющей начальную и последнюю вершины.

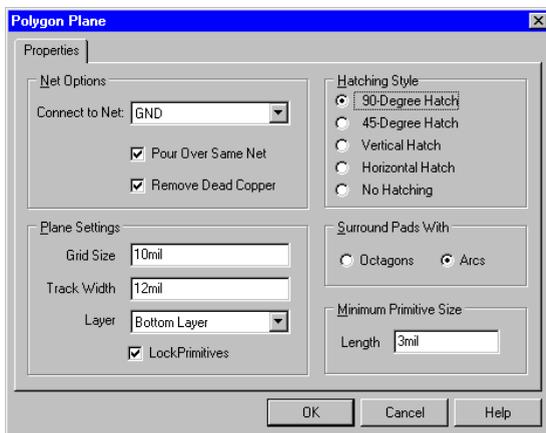


Рис. 6.23. Настройка параметров полигона.

В диалоговом окне **Place Polygon Plane** задаются следующие параметры:

Поле Net Options

Connect to net (соединить с цепью)

Если для платы существует список соединений, то любая из цепей проекта может быть выбрана в выпадающем списке **Connect To Net**. В случае соединения полигона с цепью могут быть применены две другие расположенные здесь опции.

Pour Over Same Net (объединение с проводниками этой же цепи)

Если данная опция включена, все существующие внутри полигона проводники, который является частью этой же цепи, будут объединены с полигоном.

Remove Dead Copper (удаление неиспользуемых участков металлизации)

При заливке полигона и обтекании существующих проводящих объектов неизбежно возникновение небольших участков металлизации, не соединенных ни с одной контактной площадкой, проводником или переходным отверстием назначенной цепи. Данная опция включает режим автоматического удаления таких участков.

Поле Plane Setting

Grid Size (шаг сетки)

Здесь задается шаг прорисовки линий, которыми осуществляется штриховка полигона. Для оптимально расположения этих линий желательно делать этот шаг кратным шагу выводов компонентов.

Track Width (ширина линий)

Этот параметр определяет ширину линий, которыми осуществляется штриховка полигона. В случае, когда ширина линий меньше шага сетки, то поверхность полигона будет заштрихована. Если ширина линий больше или равна шагу сетки, то полигон будет сплошным. В общем случае, для получения сплошной заливки полигона необходимо установить ширину линий немного превышающей шаг сетки.

Layer (слой)

Здесь задается слой, на котором будет размещен полигон. Полигоны могут размещаться как на сигнальных, так и на механических или любых других слоях.

Поле Hatching Style (стиль штриховки)

90 Degree Hatch

Полигон будет заштрихован горизонтальными и вертикальными линиями.

45 Degree Hatch

Полигон будет заштрихован ортогональными линиями под углом в 45°.

Vertical Hatch

Полигон будет заштрихован вертикальными линиями.

Horizontal Hatch

Полигон будет заштрихован горизонтальными линиями.

No Hatching

В этом режиме прорисовываются только внешние границы полигона, штриховка внутри него не выполняется. Данная опция полена на начальных этапах работы, когда необходимо просто обозначить наличие полигона, чтобы не снижать производительность системы, заливку которого можно будет сделать позднее.

Поле Surround Pads With (способ обтекания контактных площадок)

Обтекание контактных площадок может быть выполнено дугами или восьмиугольниками. При использовании восьмиугольников выходной файл в формате Gerber получается меньшего размера, а также увеличивается скорость вывода чертежа на фотоплоттер.

Поле Minimum Primitive Size (минимальный размер примитива)

Length

В этом поле задается минимальный размер примитивов, используемых в данном полигоне. Так как полигоны могут содержать большое количество участков линий и окружностей, используемых для реализации сглаженных кривых вокруг имеющихся объектов печатной платы, ограничение минимальной длины примитивов позволяет ускорить прокладку полигонов, перерисовку экрана, а также генерацию выходных файлов, за счет отказа от сглаживания границ полигона.

Способ соединения полигонов с контактными площадками

Способ соединения полигонов с контактными площадками определяется правилом проектирования **Polygon Connect Style** (команда меню **Design » Rules**). Соединение может быть непосредственным, а также с использованием теплового барьера. В последнем случае можно задавать число проводников, их ширину и угол наклона, а также величину зазора теплового барьера. Более подробная информация о правиле проектирования **Polygon Connect Style** приведена в разделе *Определение требований к проекту печатной платы*.

Повторная заливка полигона

Для выполнения повторной заливки полигона необходимо выбрать пункт меню **Edit » Change**, после чего необходимо выбрать нужный полигон и выполнить на нем щелчок левой кнопкой мыши. Появится диалоговое окно **Place Polygon Plane**, в котором можно отредактировать параметры полигона. После нажатия кнопки **OK** появится запрос на подтверждение повторной заливки полигона с учетом сделанных изменений, в котором следует нажать кнопку **Yes**.

Изменение границ полигона

Конфигурация границ полигона может быть изменена с помощью перемещения, добавления и удаления вершин.

Для изменения границ полигона необходимо выполнить следующие действия:

1. Выполните команду меню **Edit » Change » Polygon Vertices**, после чего в строке состояния появится подсказка **Choose a Polygon** (выберите полигон).
2. Щелкните на редактируемом полигоне левой кнопкой мыши. Линии штриховки внутри полигона исчезнут, а появится его изначальная граница.

3. На каждом сегменте границы появятся по 3 маркера-манипулятора (рис. 6.24). Щелчок на маркере-манипуляторе, расположенном в вершине полигона, позволяет перемещать ее по чертежу, а щелчок по маркеру в центре сегмента добавляет новую вершину и разбивает сегмент на две части.
4. Для удаления перемещаемой вершины используется клавиша **DELETE**.
5. Для завершения изменения полигона необходимо выполнить щелчок правой кнопкой мыши или нажать клавишу **ESC**.

После завершения изменения полигона появится запрос на подтверждение повторной заливки полигона с учетом сделанных изменений, в котором следует нажать кнопку **Yes**.

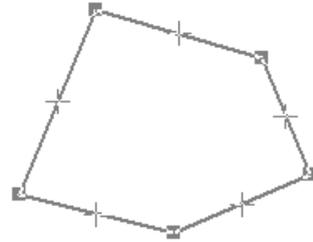


Рис. 6.24. Графическое редактирование полигонов.

Условные обозначения размеров

Размеры представляю собой сложные групповые объекты из текстовых строк и линий. Размер проставляется с помощью команды меню **Place » Dimension** после задания начальной и конечной точек. В зависимости от используемой в данный момент системы единиц в размерах будут применены мили или миллиметры.

Размещение размеров

1. Выполните команду меню **Place » Dimension**.
2. С помощью указателя мыши задайте начальную точку измерения. В строке состояния появится подсказка **Select Measure End Point** (выберите конечную точку измерения).
3. Щелчком левой кнопки мыши задайте вторую точку измеряемого объекта размера.

◆ Для разбиения условного значения размера на отдельные графические примитивы используется команда **Tools » Convert » Explode Dimension to Free Primitives**.

После всех перечисленных действий на чертеже будет размещено условное обозначение размера объекта.

Редактирование размеров

Параметры размеров, такие как высота цифр и букв, шрифт и т. д. могут быть изменены либо во время размещения (посредством нажатия клавиши **TAB**), либо позднее. Вызов диалогового окна **Change Dimension**, где осуществляется редактирование параметров размера, производится с помощью команды меню **Edit » Change** и последующего щелчка левой кнопкой мыши на нем.

Перемещение размеров

После размещения на чертеже условные обозначение размеров могут быть перемещены в другое место или заново настроены. Для перемещения или настройки размера необходимо выделить его фокусом, щелкнув на нем один раз левой кнопкой мыши. В начальной и конечной точках появятся специальные маркеры-манипуляторы.

Повторный щелчок на одном из манипуляторов позволит переместить соответствующую точку в новое место чертежа. Другая точка будет оставаться на месте, а содержимое надписи размера автоматически обновляться. При щелчке на любом другом месте размера перемещаться будет весь размер, а не только одна из его точек привязки.

Параметры размеров, задаваемые по умолчанию

Параметры размещаемого в данный момент размера могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » Dimension**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

Координатные метки

Координатные метки используются для отображения координат определенной точки чертежа печатной платы. Координатная метка включает в себя две перекрещивающиеся линии и текстовые строки координатами точки (X, Y). Координатные метки могут быть размещены на любом слое чертежа платы.

Параметры координатных меток, задаваемые по умолчанию

Параметры размещаемой в данный момент координатной метки могут быть установлены нажатием клавиши **TAB** сразу же после выполнения команды меню **Place » Coordinate**. Задаваемые по умолчанию параметры определяются на вкладке **Default** диалогового окна **Preferences**. В случае, когда опция **Permanent** вкладки **Default** выключена, все изменения, сделанные во время размещения, будут установлены как вновь принятые по умолчанию.

◆ Для разбиения координатной метки на отдельные графические примитивы используется команда **Tools » Convert » Explode Coordinate to Free Primitives**.

Размещение координатных меток

1. Выполните команду меню **Place » Coordinate**, после чего появится координатная метка, захваченная указателем мыши.
2. Щелкните левой кнопкой мыши в нужной точке.
3. Для выхода из режима размещения необходимо щелкнуть правой кнопкой мыши или нажать клавишу **ESC**.

Библиотеки топологических посадочных мест

Библиотеки системы Protel 99 SE содержат более 1600 топологических посадочных мест различных компонентов, предназначенных для поверхностного монтажа и монтажа в отверстия в плате. Топологические посадочные места создаются и редактируются в специальном редакторе PCB Library Editor, который будет описан несколько позднее в подразделе *Редактор библиотек топологических посадочных мест*.

Чем компонент отличается от топологического посадочного места?

Топологические посадочные места — это базовые графические элементы редактора чертежей печатных плат, которые хранятся в специальных библиотечных базах данных. Только в процессе разработки проекта после того, как топологическое посадочное место будет ассоциировано с одним или несколькими элементами принципиальной схемы, оно получит понятие компонента.

Где находятся библиотеки топологических посадочных мест?

В системе Protel 99 SE библиотеки топологических посадочных мест содержатся в специальных базах данных, располагающихся в папке `\Program Files\Design Explorer 99 SE\Library\Pcb`.

- ◆ К системе Protel может быть добавлена любая другая стандартная библиотека топологических посадочных мест с расширением **.LIB**.

Доступ к посадочным местам, необходимым для конструирования

Чтобы получить доступ к топологическим посадочным местам, содержащимся в той или иной библиотеке, она должна быть добавлена в список доступных библиотек редактора печатных плат. Существует два способа добавить или удалить библиотеку: нажатием кнопки **Add/Remove** на соответствующей панели или с помощью команды меню **Design » Add/Remove Library**. Появится диалоговое окно **PCB Libraries**, в котором производится подключение новых библиотек или отключение уже подключенных библиотек из списка **Selected Files**.

Библиотеки подключаются к системе только один раз, после чего их компоненты могут быть использованы для создания различных чертежей. Максимальное количе-

ство библиотек, одновременно подключенных к системе, ограничивается только ресурсами используемого компьютера.

Подключение и отключение библиотек

Для определения доступа к библиотекам компонентов используется диалоговое окно **PCB Libraries** (рис.6.25). В нижней части окна в поле **Selected Files**, показаны все подключенные в настоящее время библиотеки.

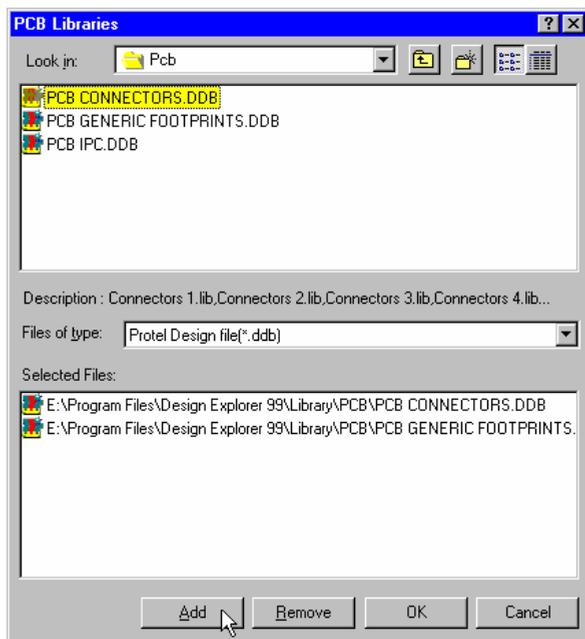


Рис. 6.25. Диалоговое окно PCB Libraries.

Для выбора папки, в которой находятся библиотеки топологических посадочных мест, используется выпадающий список **Look**, расположенный в верхней части диалогового окна. Система Protel 99 SE хранит библиотечные базы данных в папке **\Program Files\ Design Explorer 99 SE\Library\Pcb**.

Библиотеки компонентов пакета Protel 99 SE представляют собой специфические базы данных, которые можно без труда создавать и редактировать. Как правило, такие базы данных формируются из библиотек условных графических изображений для схем и топологий посадочных мест компонентов для печатных плат. Двойной щелчок левой кнопкой мыши на выбранной библиотечной базе данных добавляет ее к списку подключенных библиотек. Двойной щелчок левой кнопкой мыши на библиотеке в списке **Selected Files** позволяет отключить ее от системы.

Для того чтобы получить доступ к той или иной библиотеке компонентов, ее необходимо добавить в список **Selected Files** в диалоговом окне **PCB Libraries**.

Поиск и размещение компонентов

Для просмотра содержимого библиотечных баз данных служит панель просмотра редактора печатных плат (рис. 6.26). Для этого надо перевести панель в режим просмотра библиотек, выбрав в списке **Browse** пункт **Libraries**, и выбрать нужную библиотеку из списка ниже. В поле **Components** появится список содержащихся в библиотеке компонентов. После щелчка на любом из них, его изображение появится в браузер **MiniViewer**.

Размещение компонента на плате

Для размещения компонента на чертеже печатной платы нужно:

1. В списке **Browse** выбрать пункт **Libraries**.
2. Выбрать нужную библиотеку.
3. Выполнить на компоненте двойной щелчок левой кнопкой мыши или выбрать нужный компонент и нажать кнопку **Place**. Выбранный компонент окажется "приклеенным" к указателю мыши.
4. Для того, чтобы изменить атрибуты компонента перед размещением на чертеже, необходимо в то время, когда он "приклеен" к указателю мыши, нажать клавишу **TAB**.

Нажатие клавиш **PAGEDOWN** и **PAGEUP** позволяет изменять масштаб просмотра чертежа. Нажатие горячей клавиши **SPACEBAR** позволяет поворачивать компоненты, а клавиши **L** — переносить их на нижнюю сторону печатной платы. С помощью команды **Jump** можно переместиться в заданную точку чертежа.

5. Выполните щелчок левой кнопкой мыши для завершения размещения компонента.

Выбор и размещение компонента из редактора топологических посадочных мест

Компонент может быть выбран и размещен на чертеже печатной платы непосредственно из редактора топологических посадочных мест с помощью кнопки **Place**, расположенной на панели просмотра редактора. Выбранный компонент будет переслан

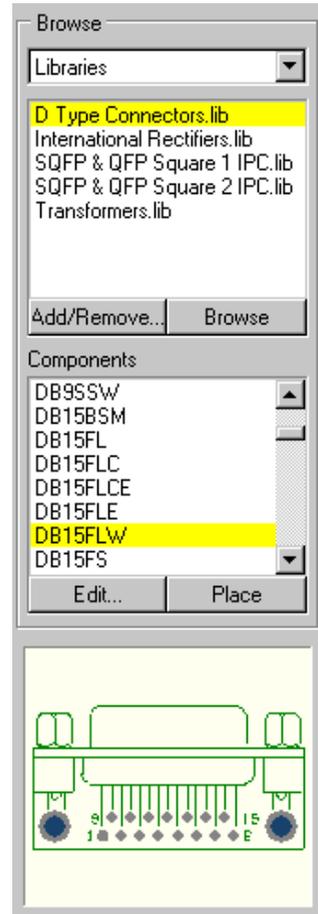


Рис. 6.26. Панель просмотра редактора печатных плат.

в активное окно редактора печатных плат, где его можно будет разместить стандартным образом.

Атрибуты компонентов печатных плат

Как только топологическое посадочное место будет расположено на чертеже печатной платы, оно становится компонентом, имеющим уникальное позиционное обозначение, например, **R3** и комментарий, например, **10K**. В редакторе печатных плат топологическое посадочное место ведет себя как один объект, а не как набор линий, кругов, дуг, контактных площадок, поэтому его можно поворачивать, отображать зеркально, перемещать и так далее. Для изменения параметров компонента необходимо выполнить команду меню **Edit » Change** или выполнить двойной щелчок левой кнопкой мыши на нужном компоненте.

Появится диалоговое окно **Change Component** с тремя вкладками **Properties**, **Designator** и **Comment**.

Вкладка Properties

На данной вкладке задаются общие атрибуты компонента (рис. 6.27).

Designator/Comment

В этих полях задаются позиционное обозначение компонента и текстовый комментарий к нему. Настройки отображения текста (шрифт, высота, ширина линий) устанавливаются на вкладках **Designator** и **Comment**.

Footprint

Текущее топологическое посадочное место компонента можно поменять на любое другое доступное в какой-либо из подключенных библиотек. Если топологическое посадочное место будет изменено, то при закрытии этого окна редактор печатных плат найдет его в указанной библиотеке и разместит его на чертеже печатной платы.

Layer

Компонент можно расположить как на верхнем (**Top**), так и на нижнем (**Bottom**) слоях печатной платы. Нужный слой выбирается из выпадающего списка **Layer**.

Изменение слоя приводит к перемещению компонента на противоположный слой. Например, при перемещении компонента с верхнего слоя на нижний слой, примитивы с верхнего слоя будут перенесены на нижний слой. Компонент зеркально отобразится относительно оси Y, а текстовые надписи компонента будут читаться на нижней сто-

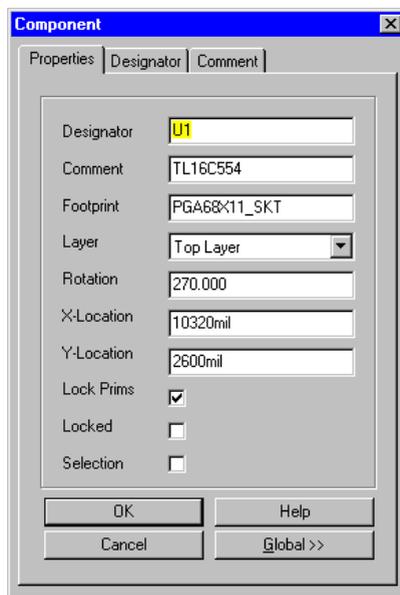


Рис. 6.27. Вкладка Properties окна редактирования атрибутов компонента.

роне платы. Одиночные контактные площадки переместятся с верхнего слоя на нижний. Изменение слоя может быть произведено для группы компонентов с помощью функций глобального редактирования.

Rotation

Здесь задается угол поворота компонента относительно сохраненного в библиотеке положения. Минимальный шаг поворота составляет 0.001° .

Lock Prims

Обычно с компонентом работают как с единым объектом, не зависимо от числа входящих в него графических примитивов. Включение данной опции связывает примитивы друг с другом. Однако если требуется индивидуальное управление примитивами или их модификация, то пользователь может разблокировать их, выключив ее, а в последствии снова заблокировать. Атрибуты контактных площадок можно изменять без разблокирования примитивов.

Locked

Атрибут **Locked** определяет, будет ли компонент зафиксирован на плате или его можно перемещать. Если эта опция включена, программа автотрассировки не сможет при необходимости передвинуть компонент на другое место. При попытке вручную передвинуть заблокированный компонент появится соответствующее предупреждение и запрос на продолжение этого действия. После ручного перемещения заблокированного компонента атрибут **Locked** остается включенным.

Selection

Компонент можно выделить или, наоборот, снять с него выделение. Использование этого атрибута помогает идентифицировать компоненты при выполнении операций глобального редактирования.

Вкладки Designator и Comment

Эти две вкладки идентичны друг другу (рис. 6.28). Они содержат атрибуты текста для отображения позиционного обозначения и комментария.

Поле Text

В этом поле записывается позиционное обозначение компонента или комментарий. Атрибут **Designator**, в отличие от атрибута **Comment**, не допускает глобального редактирования, потому что каждый компонент должен иметь уникальное обозначение. Максимальное количество символов не должно превышать 256.

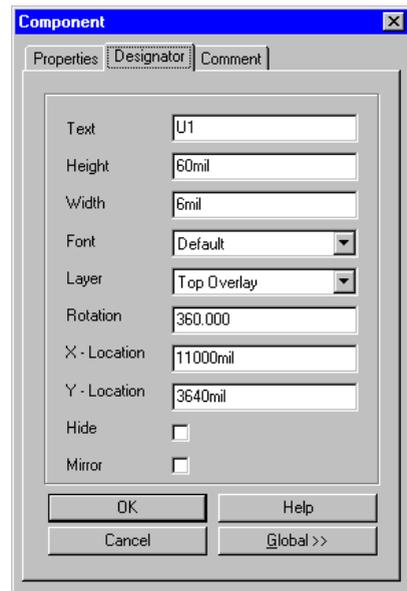


Рис. 6.28. Вкладка Designator окна редактирования атрибутов компонента.

Поле Height

Размер текста можно устанавливать в милах в диапазоне от 0.01 до 10000 мил или в мм в диапазоне от 0.254 мкм до 254 мм. Ширина символа при отображении его на экране дисплея или выводе на принтере пропорциональна его высоте. Минимальная высота символа при печати на фотоплоттере не должна быть меньше 36 мил.

Поле Width

Ширину линий прорисовки текста можно устанавливать в милах в диапазоне от 0.01 до 255 мил или в мм в диапазоне от 0.254 мкм до 6.5 мм.

Поле Font

Здесь выбирается один из трех доступных векторных шрифтов: **Default**, **Sans Serif** и **Serif**.

Поле Layer

Текст компонента может быть расположен на любом слое. Нужный слой выбирается из выпадающего списка в поле **Layer**.

Поле Rotation

Текст компонента можно передвигать или вращать независимо от самого компонента. Для перемещения текста компонента необходимо щелкнуть на нем левой кнопкой мыши и удерживая ее переместить в нужное место. Нажатием клавиши **SPACEBAR** текст можно повернуть относительно точки привязки на заранее заданный угол.

Координаты X и Y

Координаты текста на рабочем пространстве редактора печатных плат относительно текущего (относительного) начала координат.

Поле Hide

Текст компонента можно показать на экране или скрыть. Скрытый текст на печать не выводится.

Поле Mirror

Текст компонента может быть отображен зеркально, не зависимо от компонента.

- ◆ Для переноса компонента с верхнего слоя на нижний во время размещения необходимо использовать клавишу **L**. При этом корпус компонента будет зеркально перевернут, контактные площадки с верхнего слоя будут перенесены на нижний, а графические рисунки и текстовые надписи зеркально отобразятся. Не следует для этого пользоваться клавишами **X** и **Y**, так как при этом компонент просто зеркально отображается без изменения слоя размещения.

Изменение назначенного топологического посадочного места

Для того чтобы у уже размещенного на плате компонента изменить тип топологического посадочного места, необходимо выполнить двойной щелчок левой кнопкой мыши на нем, после чего появится диалоговое окно **Change Component**. Затем в поле **Footprint** на вкладке **Attributes** следует указать новый тип топологического посадочного места.

Текущее топологическое посадочное место может быть изменено на любое другое из подключенных к системе библиотек. После нажатия кнопки **OK** редактор печатных плат просмотрит все подключенные библиотеки топологических посадочных мест, найдет нужное и разместит его на чертеже печатной платы.

Топологические посадочные места компонентов свободно меняются только в том случае, если число выводов в них совпадает, что позволяет избежать появления ошибок в списке соединений. При нарушении этого условия на экран будет выведено соответствующее предупреждение, а операция замены будет прервана. Например, замена посадочного места **Dip-16** на **SMD16A** будет произведена без каких-либо затруднений, так как число выводов в обоих случаях одинаковое. Замена **Dip-16** на **TO-3** приведет к появлению предупреждения и будет прекращена. При успешной замене корпуса также будут обновлены все виртуальные линии связи, подсоединенные к соответствующим контактным площадкам.

Модификация отдельных компонентов на печатной плате

В общем случае, если необходимо внести изменения топологическое посадочное место компонента, оно редактируется в библиотеке, после чего печатная плата обновляется. При этом все старые посадочные места заменяются на новые во всех открытых в настоящий момент проектах.

Редактор печатных плат позволяет модифицировать отдельные компоненты непосредственно на печатной плате. Для этого нужно выполнить двойной щелчок левой кнопкой мыши на нужном компоненте, в появившемся диалоговом окне **Change Component** выбрать вкладку **Attributes**, выключить опцию **Lock Prims** и нажать кнопку **OK**. Теперь всеми графическими примитивами компонента можно управлять по отдельности. После модификации примитивы компонента должны быть повторно заблокированы. Отметим, что атрибуты контактных площадок можно изменять без разблокирования примитивов.

Имеется возможность добавлять новые графические примитивы к размещенному на чертеже топологическому посадочному месту компонента. Для этого сначала необходимо выключить опцию **Lock Prims**, как описано выше. Затем нужно расположить новые примитивы должным образом, выделить именно их (а не существующие примитивы компонента) и выполнить команду меню **Tools » Convert » Add Selected Primitives to Component**. Появится запрос на выбор компонента, к которому будут добавлены выделенные примитивы. Щелчок левой кнопкой мыши по нужному компоненту инициирует операцию добавления, после которой с выбранных ранее примитивов выделение будет снято. После завершения процедуры добавления рекомендуется снова включить опцию **Lock Prims**. Заметим, что изменения коснутся только выбран-

ного компонента и никак не повлияют на топологическое посадочное место в библиотеке.

Примитивы, составляющие полигон, также могут быть добавлены к посадочному месту компонента. Для этого надо разместить полигон на плате, выполнить его разбиение на отдельные примитивы командой меню **Tools » Convert » Explode Polygon to Free Primitives**, выделить и добавить их к компоненту описанным выше методом.

Включение проводников в топологическое посадочное место

Топологические посадочные места, хранящиеся в библиотеках, также могут включать в себя проводники. Существует возможность автоматического обновления имен цепей таких компонентов при передаче списка соединений из редактора схем в редактор печатных плат. Для этого необходимо при выполнении команды **Design » Update PCB**, расположенной в меню редактора принципиальных схем, в диалоговом окне **Update Design** опцию **Assign Net** установить в значение **Connected Copper**.

Присвоение имен цепей к проводникам, добавляемым к компонентам, может быть выполнено с помощью команды **Design » Netlist Manager** из меню редактора печатных плат. Для повторного назначения имен цепей контактных площадок ко всем соединенным медным примитивам (**connected copper**) используется опция **Update Free Primitives from Component Pads** из списка, вызываемого нажатием кнопки **Menu**, расположенной в верхней части диалогового окна.

Разгруппировка компонента

При необходимости размещенный на плате компонент можно обратно преобразовать в первоначальный набор графических примитивов с помощью команды меню **Tools » Convert » UnGroup Component**. После выбора компонента и подтверждения правильности выполняемых действий компонент будет преобразован в набор примитивов, а его позиционное обозначение и текстовый комментарий (если он есть) будут удалены. Отметим, что этот процесс односторонний, и повторно сгруппировать однажды разгруппированный компонент невозможно. Операция разгруппировки действует только на отдельные копии топологических посадочных мест, размещенные на чертеже печатной платы. Посадочные места, хранящиеся в библиотеке, при этом изменены не будут.

Копирование компонентов с платы в библиотеку

В системе Protel 99 SE имеется возможность копирования топологического посадочного места компонента непосредственно с чертежа платы в библиотеку. Для этого надо выбрать компонент, выполнить команду меню **Edit » Copy** и задать точку захвата. Далее необходимо открыть нужную библиотеку, перейти на панель управления редактора библиотек, выполнить щелчок правой кнопкой мыши в списке компонентов **Components** и из всплывающего меню выбрать команду **Paste**.

Редактор библиотек топологических посадочных мест

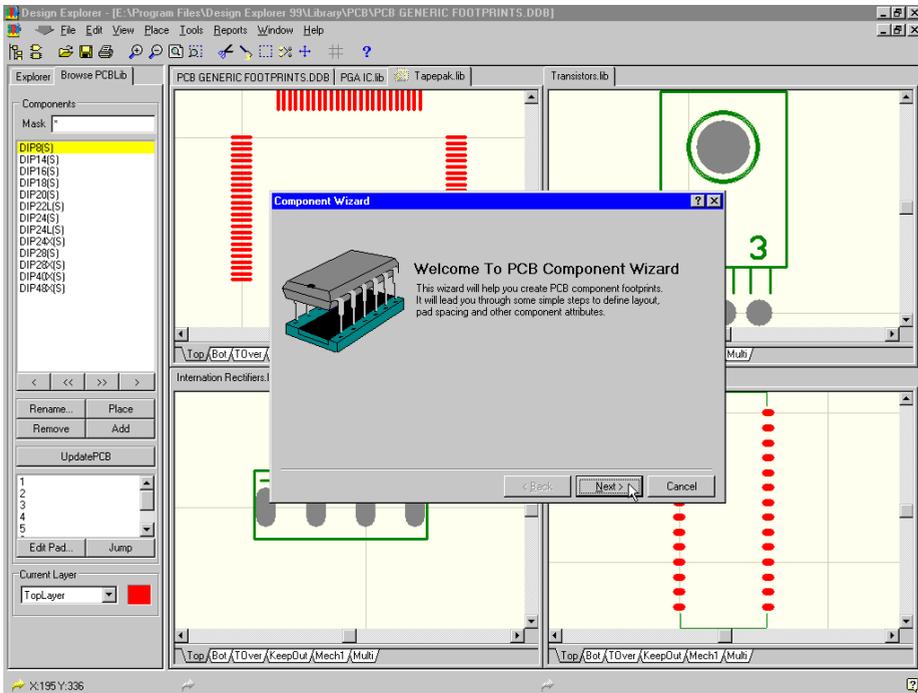


Рис. 6.29. Редактор библиотек топологических посадочных мест системы Protel 99 SE.

Редактор библиотек топологических посадочных мест является вторым редактором, представленным в PCB сервере системы Protel 99 SE. В то время как редактор чертежей печатных плат служит для размещения компонентов и проводников на чертеже, редактор библиотек топологических посадочных мест используется для создания и модифицирования посадочных мест компонентов, располагаемых на этих печатных платах (рис. 6.29). Помимо этого, он еще позволяет управлять библиотеками компонентов.

Редактор библиотек топологических посадочных мест обладает всеми функциями для создания, редактирования и размещения библиотечных посадочных мест компонентов. С его помощью можно создать собственную библиотеку компонентов. Число одновременно открытых библиотек практически не ограничено. Не существует и ограничений на количество топологических посадочных мест в одной библиотеке.

Обычно компоненты содержат не менее одной контактной площадки, число и нумерация которых совпадает с числом и нумерацией выводов соответствующих элементов, а также сегменты дорожек и или дуг, нарисованных в слое шелкографии, образующие контур компонента.

Работа с библиотеками

В системе Protel 99 SE библиотеки топологических посадочных мест, как и любые другие документы, хранятся внутри определенной базы данных проекта. В данном справочнике такие проекты называются библиотечные базы данных топологических посадочных мест (PCB Library Database).

Все такие библиотечные базы данных Protel 99 SE хранятся в папке **Program Files\Design Explorer 99 SE\Library\PCB**.

Открытие существующих библиотек

Как и все документы проекта, открываемые в среде Design Explorer, библиотеки для редактирования открываются командой **File » Open**. Каждая открытая библиотека появляется на отдельной вкладке в окне проекта.

Создание новой библиотеки

До создания новой библиотеки необходимо открыть проект, в котором нужно сохранить новую библиотеку, и в нем открыть папку, где она будет находиться.

Далее надо щелкнуть правой кнопкой мыши в окне выбранной папки и в появившемся контекстном меню выбрать опцию **New**. В появившемся диалоговом окне **New Document** следует выполнить двойной щелчок левой кнопкой мыши на иконке **PcbLib**, после чего новая библиотека будет создана.

По умолчанию новым библиотекам топологических посадочных мест в системе Protel 99 SE присваивается имя **PcbLib1**. Для того чтобы переименовать библиотеку,

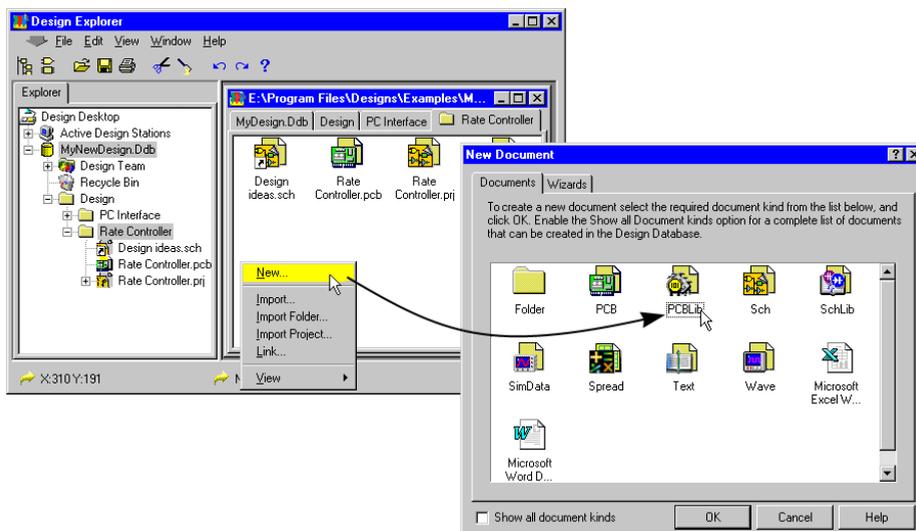


Рис. 6.30. Создание новой библиотеки.

необходимо щелкнуть на ней мышью и нажать клавишу **F2**. После этого подсветится строка, в которой можно задать новое имя библиотеки.

После ввода нового имени нажмите клавишу **ENTER**. Двойным щелчком по иконке откройте только что созданную библиотеку.

Создание топологического посадочного места с помощью мастера **Component Wizard**

Редактор библиотек топологических посадочных мест имеет очень мощную программу (мастер) по созданию компонентов. Этот мастер управляется последовательностью диалоговых окон и позволяет создать практически любое топологическое посадочное место, начиная от простейшего двухвыводного резистора и заканчивая корпусом для ПЛИС с сотнями выводов.

Запуск мастера **Component Wizard** осуществляется с помощью кнопки **Add** на панели редактора библиотек или же команды меню **Tools » New Component**.

Создание топологического посадочного места вручную

Топологические посадочные места компонентов создаются в редакторе библиотек из того же набора графических объектов, что используется в редакторе печатных плат. Здесь можно использовать угловые маркеры, реперные символы и многое другое. Типовая последовательность действий для создания топологического посадочного места вручную приведена ниже:

1. Сначала создается компонент. Для этого в редакторе библиотек необходимо открыть нужную библиотеку и выбрать команду меню **Tools » New Component**. Автоматически запустится мастер создания топологических посадочных мест. Если компонент будет создаваться вручную, нажмите кнопку **Cancel**. На экране появится чистое пустое окно с именем **PCBComponent_1**. Для редактирования имени компонента выполнить команду меню **Rename Component**. Имя компонента может иметь длину до 255 символов.
2. Затем производится размещение контактных площадок согласно заданным требованиям. Перед размещением первой контактной площадки рекомендуется выполнить команду меню **Edit » Jump » Reference** или нажать на клавиатуре горячие клавиши **J, R**, после чего указатель мыши переместится в начало координат 0,0. Определить атрибуты контактной площадки во время ее размещения можно, нажав клавишу **Tab**.

◆ Топологические посадочные места компонентов, предназначенных для поверхностного монтажа, всегда следует строить на верхнем слое. Перенос на нижний слой таких компонентов осуществляется нажатием клавиши **L**. Это крайне важно запомнить тем, кто раньше работал в системе P-CAD 2001, где эта операция выполнялась с помощью обычного зеркального отображения. В системе Protel функция зеркального отображения не переносит компонент на противоположный слой, а лишь выполняет соответствующее действие над графическими объектами.

3. После размещения контактных площадок производится прорисовка контура корпуса компонента на слое **Top Overlay** с помощью проводников. Клавиша **SPACEBAR** переключает режимы размещения **Start** и **End**. Одновременное нажатие клавиш **SHIFT+SPACEBAR** меняет режим размещения проводников.
4. Как только создание компонента будет завершено, рекомендуется сохранить текущую библиотеку. После этого можно вернуться в редактор печатных плат и разместить только что созданный компонент на чертеже платы.

Топологическое посадочное место рекомендуется строить вблизи начала координат (точки с координатами 0,0). При расположении компонента на чертеже печатной платы указатель мыши будет удерживать его именно за точку начала координат. Для изменения положения этой точки используется команда **Edit » Reference**.

В редакторе библиотек к чертежу топологического посадочного места можно добавить специальные строки **.DESIGNATOR** и **.COMMENT**. Эти надписи будут дублировать появляющиеся автоматически в редакторе печатных плат и, в отличие от них не могут быть скрыты, так как являются частью компонента.

Копирование компонентов из одной библиотеки в другую

Компоненты могут быть скопированы из одной библиотеки в другую, из библиотеки на плату и, наоборот, с платы в библиотеку. Все эти операции выполняются с помощью панели управления редактора библиотек. Компоненты здесь могут быть выбраны стандартными для системы Windows методами. После того как нужные компоненты будут выбраны, необходимо нажать правую кнопку мыши и во всплывающем меню выбрать команду **Copy**. Далее следует перейти в другую библиотеку, аналогичным образом вызвать всплывающее меню и выполнить команду **Paste**. Если выделенные компоненты необходимо вставить на плату, перейдите в редактор печатных плат и выполните команду меню **Edit » Paste**.

Обновление топологического посадочного места

После редактирования топологического посадочного места в редакторе библиотек для обновления их на всех открытых чертежах используется кнопка **Update PCB**.

Проверка компонентов

Для проверки компонентов необходимо выполнить команду меню **Reports » Component Rule Check**, после чего появится диалоговое окно **Component Rule Check**, в

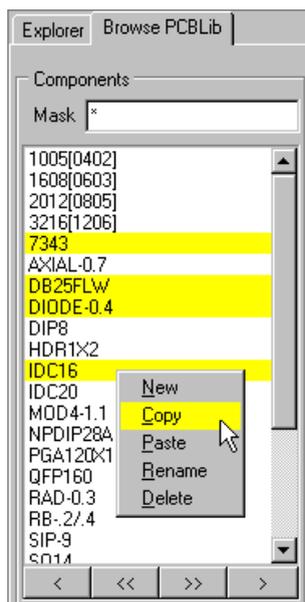


Рис. 6.31. Для перемещения компонентов между библиотеками используется всплывающее меню панели управления.

котором устанавливаются критерии проверки, например, дублированные примитивы, отсутствующие обозначения контактных площадок, несоединенные с выводами участки проводников и другие неправильно заданные параметры.

Создание библиотеки проекта

Редактор печатных плат имеет возможность создать библиотеку всех компонентов размещенных в настоящий момент на чертеже печатной платы. Для этого служит команда меню **Design » Make Project Library**, вызывающая диалоговое окно **Save Project Library**, в котором задается имя новой библиотеки. После нажатия кнопки ОК все компоненты, имеющиеся на рабочем пространстве, будут добавлены в новую библиотеку.

Описание печатной платы

Первым шагом проектирования печатной платы является задание ее границ и слоев. Несмотря на то, что законченная печатная плата представляет собой объемный (трехмерный) объект, она разрабатывается и изготавливается как набор плоских слоев, которые затем будут спрессованы друг с другом.

Процесс конструирования печатной платы заключается в размещении различных графических объектов на определенные слои платы с помощью команд меню **Place**. Физические границы платы прорисовываются линиями и дугами на специальном механическом слое. Компоненты и проводники размещаются на верхнем и нижнем сигнальных слоях и т. д.

В состав системы Protel 99 SE входит многофункциональный мастер создания печатных плат **Board Wizard**, значительно облегчающий пользователю процесс создания нового чертежа печатной платы и задания ее параметров. Мастер имеет несколько готовых шаблонов плат промышленных стандартов, а также предоставляет возможность конструктору самому создать собственный шаблон.

Если чертеж контура платы был разработан в другой системе проектирования, например, с помощью пакета AutoCAD, то его можно импортировать в редактор печатных плат с помощью специальной функции **Import DFX**.

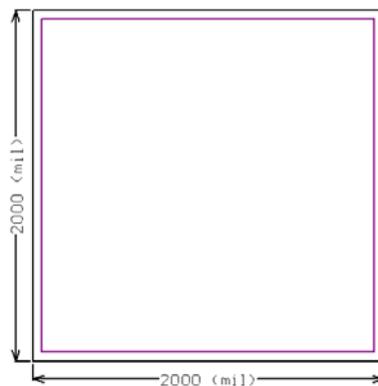


Рис. 6.32. Пример задания границ платы: внешний контур задает физические границы, а внутренний — границы размещения компонентов и проводников.

Задание физических границ платы

Состав механической информации о конструкции платы зависит от требований изготовителя и внутренних правил оформления документации. Как правило, здесь должны быть прорисованы границы платы; области, свободные от компонентов и проводником; прорези, размеры и др. Для изготовителя необходимо указать, как минимум, угловые маркеры платы, точку привязки (начало координат) и габаритные размеры. В любом случае рекомендуется уточнить эти требования у представителей производства.

Реальные физические границы печатной платы задаются проводниками, размещаемыми в одном из механических слоев платы. Включить требуемый механический слой можно с помощью команды меню **Design » Mechanical Layers**.

При задании границ платы сначала необходимо активизировать этот слой щелчком мыши по его закладке в нижней части окна редактора печатных плат (режим отобра-

жения слоев устанавливается командой меню **Design » Options**). Затем при помощи линий (команда меню **Place » Line**) необходимо обозначить границу платы. Подробную информацию о технике размещения проводников можно получить в подразделе *Линии* раздела *Объекты редактора печатных плат*. Как правило, граница платы смещается относительно абсолютного начала координат на 1 дюйм вправо и вверх.

◆ Содержимое механических слоев может быть без труда добавлено к любому слою при генерации выходных файлов. Эта возможность облегчает прорисовку размеров и физических границ платы при выполнении любой распечатки.

Определение границ размещения и трассировки

Внешние границы области размещения компонентов и трассировки проводников определяются посредством создания контура на слое **Keep Out**. Обычно эти границы определяются внутри физических границ платы с помощью установки небольшого отступа для того, чтобы проводники и компоненты не располагались слишком близко к краям платы. Сам контур прорисовывается с помощью контурных линий и дуг. Все компоненты и проводники могут быть размещены только в пределах установленных таким образом границ, которые в дальнейшем будут учитываться программами проверки правил проектирования (**Design Rule Checker**), автоматического размещения и автотрассировки.

Также имеется возможность зарезервировать любые другие области в механических границах платы, в которых компоненты и/или проводники не должны размещаться. Такие области могут использоваться для установки крепежных деталей и профилирования. Описываются они посредством размещения в слое **Keep Out** графических примитивов, например, линий, дуг и залитых объектов.

Основное правило использования слоя **Keep Out** заключается в том, что при трассировке проводников на сигнальных слоях они никогда не будут пересекать графические объекты, расположенные в слое **Keep Out**.

Области **Keep Out** могут быть заданы и для отдельных слоев. Для этого на нужный слой необходимо разместить рисунок с помощью команды **Place » Keepout**.

Задание стека слоев

После задания физических границ платы необходимо определить количество электрических слоев, используемых при изготовлении настоящей платы. Набор таких слоев называется стеком слоев.

Существует два типа электрических слоев — сигнальные слои, в которых проходят проводники, и внутренние слои питания и заземления, состоящие из сплошной ме-

таллизированной поверхности и предназначенные для подведения значительного тока к компонентам с большой потребляемой мощностью.

Стек слоев задается в диалоговом окне **Layer Stack Manager** (рис. 6.33), вызываемом командой меню **Design » Layer Stack Manager**. Рисунок в центре этого окна показывает текущий стек слоев, который по умолчанию представляет собой простую двухстороннюю плату. Новый слой добавляется под текущий с помощью кнопок **Add Layer** и **Add Plane**. Двойной щелчок левой кнопкой мыши на слое позволяет вызвать окно, в котором можно изменить его параметры.

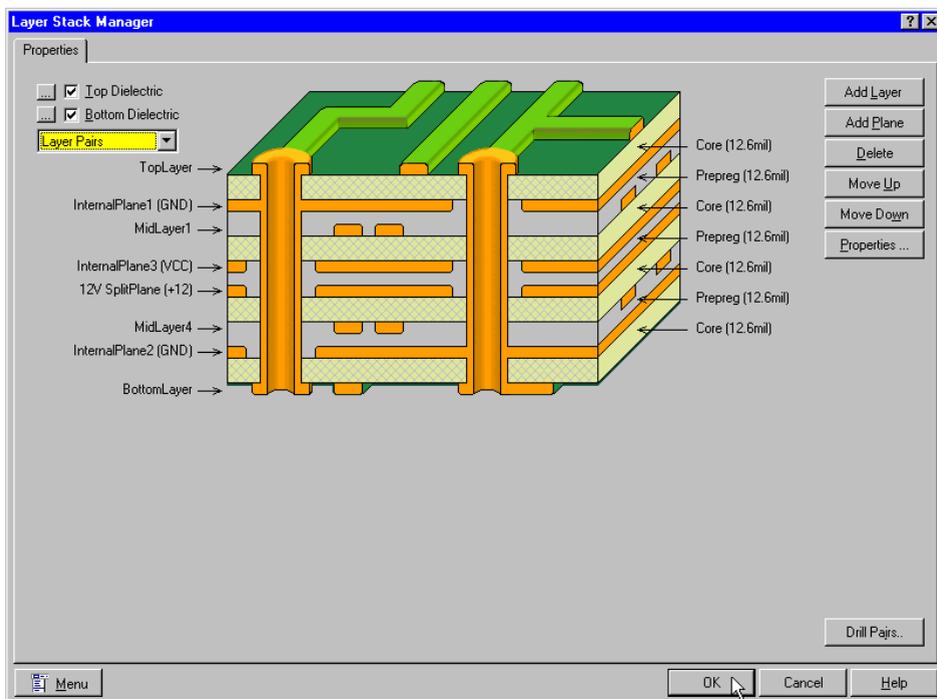


Рис. 6.33. Электрические слои определяются в диалоговом окне Layer Stack Manager.

С помощью кнопки меню можно загрузить один из заранее определенных примеров стеков слоев. Каждый из этих примеров может легко изменен. Перемещение слоев по стеку осуществляется с помощью кнопок **Move Up** и **Move Down**. Новый слой может быть добавлен в любой момент работы над проектом.

В системе Protel 99 SE может быть использовано до 32-х сигнальных слоев (верхний, нижний и 30 внутренних) и до 16-ти внутренних слое питания и заземления. Режим отображения слоев задается в диалоговом окне **Document Options**, вызываемом командой меню **Design » Options**.

Выбор стиля расположения проводящих и изолирующих слоев

Стек слоев состоит из проводящих и изолирующих слоев. При изготовлении печатных плат используется два вида изоляционных материалов: стеклотекстолитовые основания (core) и препреговые слои (композиционный пластик).

Стиль расположения слоев указывает порядок изолирующих слоев в стеке. Здесь могут быть использованы три стиля: парные слои, внутренние парные слои и нарастаемые слои. Смена стиля расположения слоев приводит к изменению варианта чередования стеклотекстолитовых листов и слоев композиционного пластика в стеке слоев.

Выбор предпочтительного стиля осуществляется в левом верхнем углу диалогового окна **Layer Stack Manager**. Задание стиля расположения слоев необходимо лишь в том случае, когда конструктор планирует использовать в своем проекте глухие и скрытые переходные отверстия, а также при анализе целостности сигналов. Прежде чем применять эти виды переходных отверстий в своей печатной плате конструктору следует проконсультироваться с представителями производства о корректности выбранного стиля расположения слоев в стеке.

Задание параметров слоев

Диалоговое окно **Layer Stack Manager** позволяет добавить в стек один из трех типов слоев: сигнальный слой, внутренний слой питания или заземления и изолирующий слой. При намерении разработчика провести анализ целостности сигнала (команда меню **Tools » Signal Integrity**) необходимо корректно указать следующую информацию:

Для сигнальных слоев:

- **Name** – задаваемое пользователем имя слоя;
- **Copper thickness** – толщина слоя металлизации, необходимая для анализа целостности сигналов.

Для внутренних слоев питания и заземления:

- **Name** – задаваемое пользователем имя слоя;
- **Copper thickness** – толщина слоя металлизации, необходимая для анализа целостности сигналов.

Для подложек и изолирующих слоев:

- **Material** – тип используемого материала;
- **Thickness** – толщина диэлектрического слоя, необходимая для анализа целостности сигналов.
- **Dielectric constant** – относительная диэлектрическая проницаемость материала, необходимая для анализа целостности сигналов.

Задание пар слоев для сверления отверстий

При формировании стека слоев конструктору необходимо определить пары слоев для сверления отверстий (**drill-pairs**). Термин **drill-pairs** относится к двум слоям, участвующих в сверлении (начальный и конечный слой). Если на плате не применяются глухие и скрытые переходные отверстия, то в проекте присутствует только одна пара

слоев сверления, состоящая из верхнего и нижнего слоя. Эта пара слоев устанавливается по умолчанию и не подлежит ни удалению, ни модификации.

Пары слоев сверления задаются в диалоговом окне **Drill-Pair Manager**, которое вызывается нажатием кнопки **Drill-Pair** в диалоговом окне **Layer Stack Manager**.

При использовании в проекте глухих и скрытых переходных отверстий, пары слоев сверления должны быть определены с учетом используемого стиля стека слоев в строгом соответствии с требованиями представителей производства.

Мастер создания печатных плат Board Wizard

Мастер создания печатных плат **Board Wizard** позволяет конструктору выбирать шаблон из большого числа готовых шаблонов плат промышленных стандартов. Шаблон может включать основную надпись, координатные метки, опорные линии, размеры и топологии стандартных разъемов. С помощью мастера также задаются текстовые строки основной надписи, количество сигнальных слоев и технология прорисовки проводников и переходных отверстий. Мастер запускается из вкладки **Wizards** диалогового окна **New Document**.

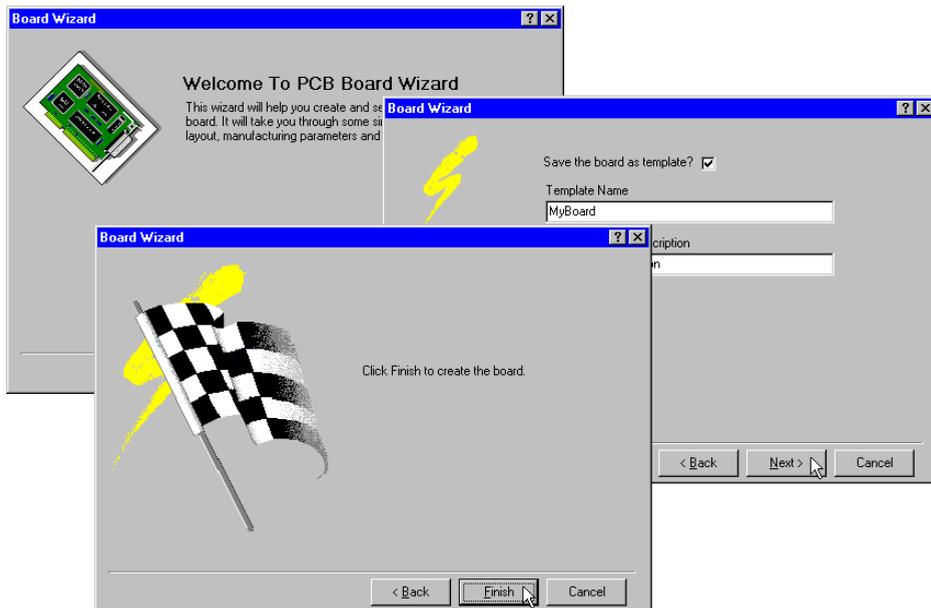


Рис. 6.34. Мастер PCB Board Wizard значительно упрощает процедуру создания новой платы.

В мастере существует возможность создания платы произвольного вида. В этом случае необходимо определить внешние размеры и форму платы, а также штампованные отверстия в ней. На последней странице мастера имеются средства для сохранения

только что введенных параметров в качестве шаблона. После сохранения при следующих запусках мастера вновь созданный шаблон появится в списке ранее определенных шаблонов.

Если пользователь включил опцию **Save the Board as a Template**, то существующий проект платы будет сохранен в специальной базе данных проекта `\Program Files\Design Explorer 99 SE\System\Templates.Ddb`. При необходимости добавления дополнительной информации в этот файл его можно редактировать. Для добавления собственных графических рисунков в мастер необходимо создать графический файл с размером рисунка 64x32 пикселей (и с таким же именем, как и у файла шаблона) и сохранить его в папке `\Boards` базы данных `Templates.Ddb`.

Передача данных из редактора принципиальных схем

После определения границ платы и отверстий крепления все готово для передачи данных с этапа разработки принципиальной схемы на этап разработки печатной платы. Эта процедура подробно описана в разделе *Передача информации о схеме в редактор печатных плат* главы *Разработка принципиальных схем*.

Определение требований к проекту печатной платы

На сегодняшний момент проектирование печатной платы — это не только простое размещение проводников для создания печатных соединений. Высокоскоростные логические схемы в сочетании с все более усложняющимися технологиями упаковки кристаллов интегральных схем в корпуса, которые постоянно уменьшаются в размерах, накладывают новые требования на проектирование печатных плат. По этим причинам стало невозможно удовлетворить все требования лишь посредством задания расстояний между проводниками, контактными площадками и переходными отверстиями. В современных проектах специфические требования могут накладываться на отдельные цепи, компоненты или части плат, а также должны учитываться такие факторы, как перекрестные искажения, отражения сигналов и длины цепей. Для успешного решения сегодняшних задач в этой области необходимо наличие инструмента, который может быть сконфигурирован для мониторинга и проверки всех перечисленных требований. В системе Protel 99 SE эта задача решается посредством определения правил проектирования.

Редактор печатных плат включает большое количество правил проектирования, определяющих зазоры, геометрию объектов, размещение, параллельность, импеданс, целостность сигналов, приоритеты при трассировке и топологию трассировки.

Каждое правило имеет область действия (**Rule Scope**), которая определяет, где это правило применяется. Область действия правила проектирования точно определяет, к каким объектам платы должно быть применено данное правило и может составлять как всю плату, так всего одну контактную площадку. Возможно задание нескольких правил проектирования одного типа, в том числе и с перекрывающимися областями действия. Например, одно значение зазоров между проводниками может быть опре-

делено для всей платы, а другое — всего для одной цепи. Все правила строятся по иерархическому принципу, определяющему их приоритет.

Что такое правило проектирования?

Разработка печатной платы начинается с размещения компонентов, проводников, переходных отверстий и других объектов. Все эти объекты должны быть размещены на чертеже относительно друг друга согласно некоторым правилам. Компоненты не должны перекрываться, различные цепи не должны соединяться и пересекаться, цепи питания и заземления должны избегать проникновения в них сигналов и помех, разные цепи должны иметь разную ширину, некоторые цепи должны иметь одинаковую длину и так далее.

Редактор печатных плат имеет возможность автоматически отслеживать нарушение перечисленных требований, что позволяет проектировщику заниматься непосредственно задачами разработки, не отвлекаясь на другие проблемы. Для этого необходимо указать редактору печатных плат набор таких требований, что можно сделать с помощью установки правил проектирования. Редактор печатных плат следит за размещением объектов на чертеже и, как только происходит нарушение установленных

◆ Редактор печатных плат позволяет создавать классы (наборы) объектов, что является очень удобным средством для настройки и управления правилами проектирования. Более подробная информация по созданию классов цепей, компонентов или маршрутов приведена в подразделе *Классы объектов* данной главы.

правил, размещаемый объект подсвечивается.

Определение правил проектирования

Где определяются правила проектирования?

Установка и редактирование правил проектирования производится в диалоговом окне **Design Rules** (рис. 6.35), которое вызывается с помощью команды меню **Design » Rules**.

Все доступные в редакторе печатных плат правила проектирования делятся на шесть групп, каждой из которых в диалоговом окне **Design Rules** выделена отдельная вкладка. В левой части каждой вкладки имеется список, состоящий из набора существующих правил. Далее за списком следует описание выбранного в данный момент правила проектирования.

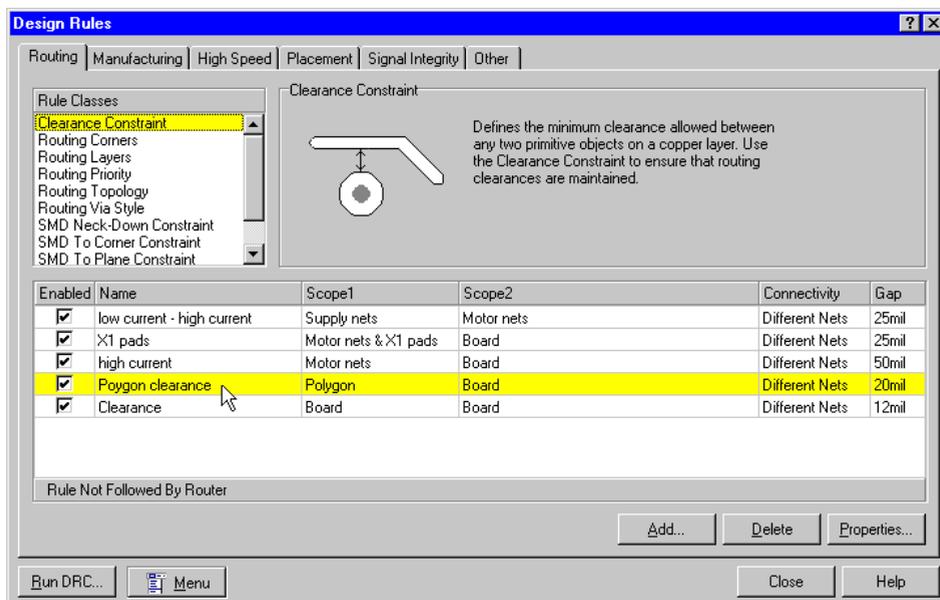


Рис. 6.35. Правила проектирования устанавливаются и редактируются в диалоговом окне Design Rules.

В нижней части окна отображается список всех определенных для данной платы правил проектирования, включая задаваемые по умолчанию. Обратите внимание, что каждое установленное правило может быть выключено или включено, что позволяет исключить его из процесса проверки без удаления из проекта. Расположение правил проектирования в списке определяется их приоритетом. Чем выше правило в списке, тем выше его приоритет.

Добавление правила проектирования

Для добавления требуемого правила в список учитываемых необходимо найти его на вкладках диалогового окна **Design Rules**, выбрать и затем нажать кнопку **Add** или выполнить на нем двойной щелчок левой кнопкой мыши. В результате для выбранного правила на экран будет выдано соответствующее ему диалоговое окно, в котором можно сконфигурировать это правило и установить его область действия (рис. 6.36).

Каждому новому правилу проектированию присваивается имя по умолчанию, которое без труда можно изменить в поле **Rule Name**. Оригинальное имя помогает идентифицировать правила на панели управления редактором печатных плат.

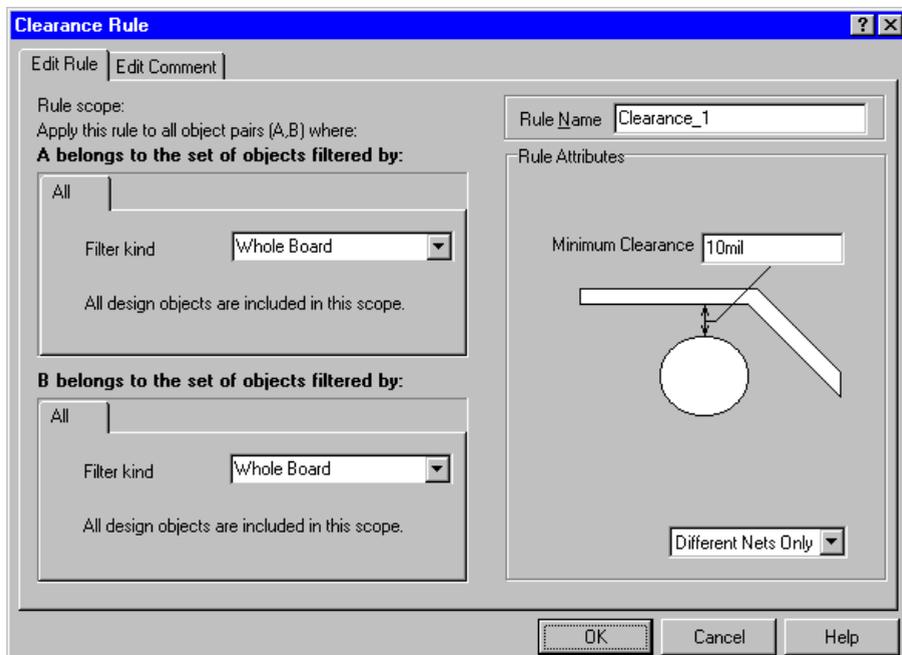


Рис. 6.36. Каждое правило конфигурируется в собственном диалоговом окне Rule.

Что такое область действия правила проектирования?

Область действия правила проектирования определяет множество объектов, на которое распространяется данная реализация этого правила. Указанием области действия можно установить распространение данного правила либо на всю плату, либо на отдельную цепь, компонент или контактную площадку.

Описание областей действия правил проектирования

Ниже приведено описание возможных областей действия правил проектирования.

Whole Board (вся плата)

Данная область имеет самый низкий приоритет. В этом случае правило применяется ко всем без исключения объектам на плате.

Layer (слой)

Данная область определяет все объекты на указанном слое, но сами объекты здесь не указываются. Примером может служить задание различных параметров окон в масках для нанесения паяльной пасты на нижнем и верхнем слоях платы.

Object Kind (тип объектов)

В этом случае правило применяется к определенному типу объектов. Примером может служить случаи, когда зазор между проводниками и контактными площадками должен отличаться от зазора между разными проводниками.

Footprint (посадочное место)

В этом случае правило применяется только к тем компонентам, которые имеют заданное топологическое посадочное место. Преимущество такой области действия в том, что если в проект добавляется компонент, имеющий уже используемое топологическое посадочное место, и для этого посадочного места уже задано правило проектирования, то этот компонент автоматически попадает под его установки. Примером могут служить окна в масках для пайки компонентов с различным конструктивным исполнением.

Component Class (класс компонентов)

При таком задании области действия правило распространяется на все компоненты заданного класса. Примером могут служить группы размещаемых компонентов, использующих правило **Room Definition** или компоненты со специфической ориентацией, использующих правило **Component Orientations**.

Component (компонент)

В этом случае правило распространяется только на указанный компонент. Примером могут служить индивидуально назначаемые правила размещения и создания трафаретов для пайки для разных компонентов.

Net Class (класс цепей)

Данная область действия охватывает все цепи указанного класса, причем под понятием цепь подразумеваются и проводники, и контактные площадки, и дуги, и области заливки, и переходные отверстия ее образующие. Примером могут служить правила, определяющие ширины проводников и зазоры между ними, а также правила проектирования для высокоскоростных плат и контроля целостности сигналов.

Net (цепь)

Данная область действия охватывает только указанную цепь, причем под понятием цепь подразумеваются и проводники, и контактные площадки, и дуги, и области заливки, и переходные отверстия ее образующие. Примером могут служить правила, определяющие ширины проводников и зазоры между ними, а также правила проектирования для высокоскоростных плат и контроля целостности сигналов.

From-To Class (класс маршрутов)

В этом случае правило распространяется на заданный класс определенных пользователем участков цепи от одной контактной площадки к другой (команда меню **Design » From-To Editor**), причем в состав маршрута могут входить и проводники, и контактные площадки, и дуги, и области заливки, и переходные отверстия. Примером могут служить правила, определяющие ширины проводников и зазоры между ними, а также правила проектирования для высокоскоростных плат и контроля целостности сигналов.

From-To (маршрут)

В этом случае правило распространяется только на указанный определенный пользователем участок цепи от одной контактной площадки к другой (команда меню **Design » From-To Editor**), причем в состав маршрута могут входить и проводники, и контактные площадки, и дуги, и области заливки, и переходные отверстия. Примером могут служить правила, определяющие ширины проводников и зазоры между ними, а также правила проектирования для высокоскоростных плат и контроля целостности сигналов.

Pad Class (класс контактных площадок)

В этом случае правило распространяется только контактные площадки указанного класса. Примером может служить размер окон в маске для пайки, стиль соединения контактных площадок с полигоном или внутренним слоем питания, а также ограничения размера отверстий.

Pad Specification (специфический набор контактных площадок)

В этом случае правило распространяется только на те контактные площадки, которые удовлетворяют некоторому критерию. Для задания этого критерия необходимо нажать кнопку **Specification** в диалоговом окне **Rule**, после чего появится окно **Pad Specification**. Здесь необходимо выключить неиспользуемые параметры, включить нужные и правильно их сконфигурировать. Примером может служить размер окон в маске для пайки, стиль соединения контактных площадок с полигоном или внутренним слоем питания, а также ограничения размера отверстий.

Via Specification (специфический набор переходных отверстий)

В этом случае правило распространяется только на те переходные отверстия, которые удовлетворяют некоторому критерию. Для задания этого критерия необходимо нажать кнопку **Specification** в диалоговом окне **Rule**, после чего появится окно **Via Specification**. Здесь необходимо выключить неиспользуемые параметры, включить нужные и правильно их сконфигурировать. Примером может служить размер окон в маске для пайки, стиль соединения контактных площадок с полигоном, а также ограничения размера отверстий. Другим примером может служить предотвращение соединения с внутренним слоем питания. Чуть позже такой пример будет рассмотрен.

Footprint Pad (контактная площадка данного посадочного места)

Данная область действия охватывает только указанную контактную площадку (или площадки — при использовании символов-заменителей) определенного топологического посадочного места.

Pad (контактная площадка)

Данная область действия охватывает только указанную контактную площадку. Примером может служить размер окна в маске для пайки, стиль соединения контактных площадок с полигоном и ограничение размера отверстия.

Region (область платы)

В этом случае правило проектирования применяется к некоторой области на плате. Такие участки плат можно определить с помощью кнопки **Define**, расположенной в диалоговом окне **Rule** при задании области действия. Объект считается заключенным

в такую область, если хотя бы один из его примитивов попадает в нее. Примером может служить стиль соединения контактных площадок с полигоном или внутренним слоем питания, а также ограничение ширины проводников.

Унарные и бинарные правила, установка их области действия

Существует два типа правил проектирования — унарные и бинарные правила. Унарные правила воздействуют на один объект или на каждый отдельный объект в группе. Бинарные правила действуют между двумя объектами или между объектом из одного набора и объектом из другого.

Примером унарного правила является параметр **Solder Mask Expansion**. Это правило применяется индивидуально к каждой контактной площадке, которая попадает в область действия правила. Примером бинарного правила является ограничение на размер зазора, которое действует между любым медным объектом из первого набора и любым медным объектом из второго набора, поскольку определяются две области действия. При конфигурировании унарного правила устанавливается одна область действия, при настройке бинарного — две.

Использование классов и символов-заменителей для задания области действия правил проектирования

Существует много способов использования правил проектирования для удовлетворения требований к разрабатываемой плате. Для того, чтобы сохранить контроль за установкой правил проектирования нужно помнить несколько полезные советов по их настройке. Наборы объектов могут определяться двумя способами.

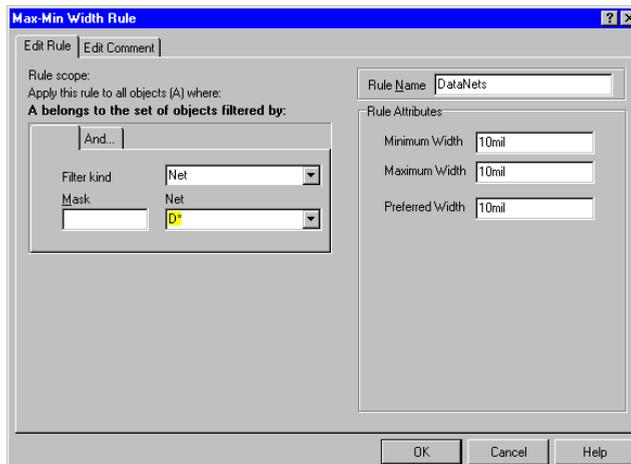


Рис. 6.37. Использование символов-заменителей для идентификации набора цепей.

Во-первых, существует возможность создавать пользовательские наборы объектов, включающие классы цепей, классы компонентов и классы маршрутов, которые могут быть использованы при задании установке действия правил проектирования. Команда меню **Design** » **Classes** предназначена для создания нового класса. Процесс создания класса описан в этой главе в подразделе *Классы объектов*.

Во-вторых, для определения набора объектов могут быть использованы символы-заменители. Поддерживаются оба общепринятых символа-заменителя: ? — для любого одиночного символа и * — для любого количества символов. В примере, изображенном на рисунке 6.37 ширина всех проводников цепей **Data** (а точнее всех цепей, начинающихся с буквы **D**) должна быть равна 10 тысячным дюйма (10 mil). Данное правило определяет, что все цепи, имена которых начинаются с буквы **D**, будут иметь ширину, равную 10 тысячным дюйма.

Составная область действия правил проектирования

Иногда возникает необходимость указать набор объектов, который является подмножеством набора, на который распространяется одиночное правило. Примером может служить цепь, проводники которой должны иметь определенную ширину на всех слоях, за исключением одного, где ширина должна быть другой.

Для установки более специфических правил проектирования имеется возможность создавать составные области действия правил. Такие области формируются с помощью логической операции И, примененной к нескольким областям (рис. 6.38). Примеры использования составных областей действия правил приведены в подразделе *Примеры использования правил проектирования* данной главы.

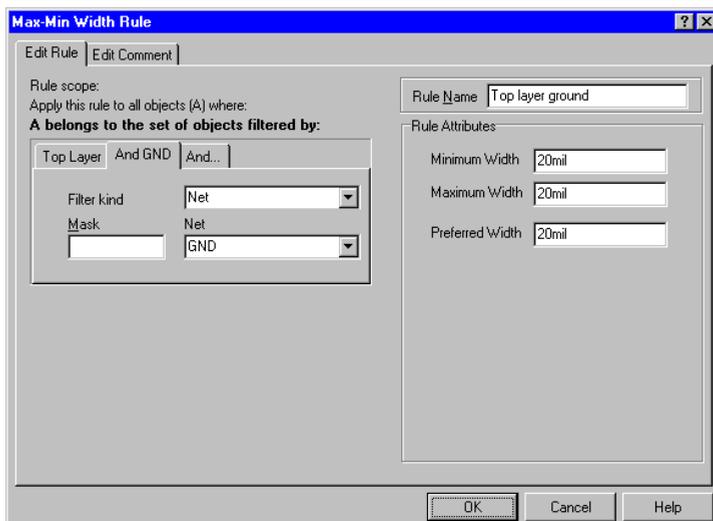


Рис. 6.38. Задание составной области действия правила проектирования позволяет обеспечить ширину проводников цепи GND на верхнем слое равной 20 мил.

При конструировании составной области действия важно понимать, на какие объекты должно воздействовать данное правило. Необходимо указывать абсолютно все критерии отбора объектов. Если это не будет сделано, правило проектирования применяться не будет. Корректность установки области действия проверяется с помощью команды **Applicable Rules** из всплывающего меню, вызываемого нажатием правой кнопки мыши. Подробности по использованию этой функции приведены в разделе *Работа с правилами проектирования* данной главы.

Пример использования составной области действия правила проектирования приведен в разделе *Примеры использования правил проектирования*.

Установка нескольких правил одного типа и их приоритеты

Область действия правила позволяет точно определить объекты, к которым оно должно быть применено. Например, правило ограничения зазора можно распространить на цепь, маршрут или отдельную контактную площадку.

После установки области действия некоторого правила проектирования можно перекрыть эту область другим правилом такого же типа.

Каждое правило можно устанавливать столько раз, сколько потребуется. Например, можно установить правило **Solder Mask Expansion** на всю плату, затем, не отменяя первой установки, на отдельный компонент и третий раз на отдельную контактную площадку этого компонента. Для того чтобы редактор печатных плат знал, какое из этих трех правил необходимо применить к указанной контактной площадке, существует механизм определения приоритетов для правил одного и того же типа, но которые имеют разные области действия. Далее перечислены области действия правил в порядке убывания приоритета:

- Область платы (высший приоритет);
- Отдельная контактная площадка;
- Контактная площадка посадочного места;
- Специфический набор переходных отверстий;
- Специфический набор контактных площадок;
- Класс контактных площадок;
- Маршрут;
- Класс маршрутов;
- Цепь;
- Класс цепей;
- Компонент;
- Класс компонентов;
- Топологическое посадочное место;
- Тип объектов;

- Слой;
- Вся плата (низший приоритет).

Разработчику не обязательно помнить описанные приоритеты, так как в списке, с помощью которого устанавливаются области действия правил, эти области перечислены в порядке убывания приоритетов, то есть от высшего к низшему.

Это позволяет при установке правил использовать стратегию "от общего к частному", то есть сначала устанавливаются наиболее общие правила, распространяющиеся на всю плату, затем создаются конкретные правила, область действия которых имеет более высокий приоритет. Редактор печатных плат подвергает анализу установленные для объекта правила и применяет к нему только те, которые имеют наивысший приоритет для данного объекта.

Конфликты из-за дублированных правил

Когда объект покрывается несколькими правилами с одинаковыми областями действия (например, двумя правилами **Solder Mask Expansion** с одинаковыми и пересекающимися областями действия) возникает конфликтная ситуация. Редактор печатных плат имеет предварительно установленную стратегию разрешения всех возможных конфликтных ситуаций. В основе используемого здесь подхода лежит принцип ошибки без ущерба надежности. То, как этот принцип интерпретируется в каждой конкретной ситуации, приводится в описаниях правил проектирования.

Применение правил проектирования

Как говорилось ранее, в редакторе печатных плат может быть установлено несколько правил проектирования одного типа. Однако, к каждому отдельному объекту будет применено только одно правило данного типа, которое будет выбрано автоматически согласно существующей системе приоритетов. Стратегия задания правил проектирования будет приведена в подразделе *Работа с правилами проектирования*.

Следует также помнить, что не все правила проверяются на том или ином этапе работы с проектом. Одни из них используются программой автотрассировки (например, **Ruting Layer**), другие используются при генерации выходных файлов для производства (например, **Solder Mask Expansion**), третьи проверяются модулем DRC. Ниже приведен обзор подобных ситуаций.

Проверка правил проектирования в интерактивном режиме (DRC)

Нарушение правила проектирования индицируется во время размещения сразу же после обнаружения этого нарушения. В этом случае объект приводится установленным в текущий момент цветом. Выключение данной функции осуществляется на вкладке **Options** диалогового окна **Preferences**.

Пакетная проверка

С помощью команды меню **Tools » Design Rule Check** вызывается диалоговое окно **Design Rule Check**, где можно выбрать типы правил, которые необходимо протестировать. После нажатия кнопки **OK** все реализации правил указанных типов будут протестированы.

Заметим, что можно устанавливать количество выдаваемых нарушений, что используется для облегчения обработки выдаваемых сообщений.

Во время выполнения автоматизированных операций

Некоторые правила проверяются во время автоматизированных операций, которые включают заливку полигонов, автотрассировку, автоматическое размещение и генерацию выходных результатов. Например, правило **Solder Mask Expansion** может отслеживаться во время генерации выходного файла, а правило описания стиля трассировки переходных отверстий **Via Style** — во время автотрассировки.

Экспортирование проекта

В пакет включены некоторые правила проектирования для поддержки функций программы автоматической трассировки SPECCTRA. Требования, устанавливаемые посредством таких правил экспортируются вместе с проектом.

Работа с правилами проектирования

Правила проектирования представляют собой базовые компоненты описания проекта и управления автоматизированными функциями редактора печатных плат. Как правило, на разных этапах разработки платы проектировщик акцентируется на разных технических требованиях. Например, во время размещения компонентов должно контролироваться минимальное расстояние между компонентами, во время трассировки учитываются правила, задающие зазоры между проводниками. В состав редактора плат входит несколько мощных функций контроля правил проектирования.

Контроль правил проектирования во время работы

Многие из правил проектирования могут контролироваться во время работы в редакторе плат. Такой режим называется интерактивной проверкой правил проектирования (on-line DRC) и включается на вкладке **On-line** диалогового окна **Design Rule Check**, вызываемого командой меню **Tools » Design Rule Check**. За исключением правила **Un-Routed Net Constraint**, все остальные предупреждения о нарушении правил проектирования, генерируемые время интерактивной проверке, немедленно отображаются на экране компьютера зеленым цветом (цвет по умолчанию для ошибок DRC). Все нарушения также отображаются в списке на панели управления редактора, для чего надо в списке Browse установить режим просмотра **Violations**.

Дополнительная информация по контролю правил проектирования приведена в разделе *Верификация проекта печатной платы*.

Управление правилами через панель управления

Установленные в проекте правила проектирования отображаются в списке на панели управления при выбранном в списке **Browse** режиме **Rules** (рис. 6.39). Для редактирования выделенного правила используется расположенная здесь кнопка **Edit**. Кнопка **Select** позволяет выбрать все объекты, попадающие под действие данного правила, а кнопка **Highlight** — подсветить такие объекты. Отметим, что функции **Select** и **Highlight** никак не влияют на состояние данных объектов, а только выделяют их на чертеже платы.

Выключение правил проектирования

Все установленные в проекте правила проектирования могут быть выключены в диалоговом окне **Design Rules**. Это действие аналогично удалению правила, но при этом оно остается в проекте, его параметры не теряются. При необходимости такое правило можно снова включить. Выключенные правила проектирования отображаются в списке на панели управления, но имена их будут перечеркнуты.

Импорт и экспорт наборов правил проектирования

Наборы правил проектирования могут быть импортированы в проект или экспортированы из него, позволяет хранить и использовать по мере надобности некоторые из них. Для этого используется кнопка **Menu**, расположенная в нижней части диалогового окна **Design Rules**. При импорте правила могут быть записаны поверх (**Overwrite**) и добавлены (**Add**) к уже существующим правилам проектирования. Нужный режим импорта выбирается в диалоговом окне **Import Options**.

Просмотр правил, приложенных к объектам

Пользователь может без труда проверить, какие из установленных правил проектирования приложены к тому или иному объекту. Для этого используются команды меню **Applicable Unary Rule** и **Applicable Binary Rule**, после запуска которых в строке состояния появится запрос на выбор одного (для унарных) и двух (для бинарных) правил.

Появится диалоговое окно **Applicable Rules**, в котором будут приведены все правила проектирования, приложенные к указанным объектам (рис. 6.40). Отметим, что здесь будут приведены абсолютно все приложенные правила одинакового типа, даже те, которые не учитываются в силу их низкого приоритета. Уровень приоритета обозна-

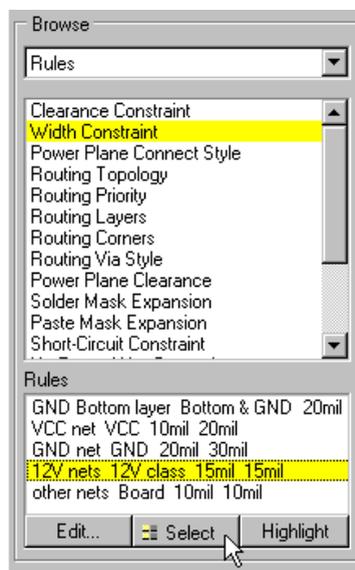


Рис. 6.39. Работа с правилами проектирования через панель управления.

чается значком слева от правила: "галочка" указывает на наивысший приоритет, а "крестик" — на низший (то есть, это правило не используется).

Кроме проверки прилагаемых к объекту правил, имеется возможность проверить все объекты, к которым некоторое правило прилагается. Для этого используется панель управления редактором печатных плат, где в списке **Browse** надо выбрать режим **Rules** и выделить или подсветить контролируемые объекты с помощью кнопок **Select** и **Highlight** соответственно.

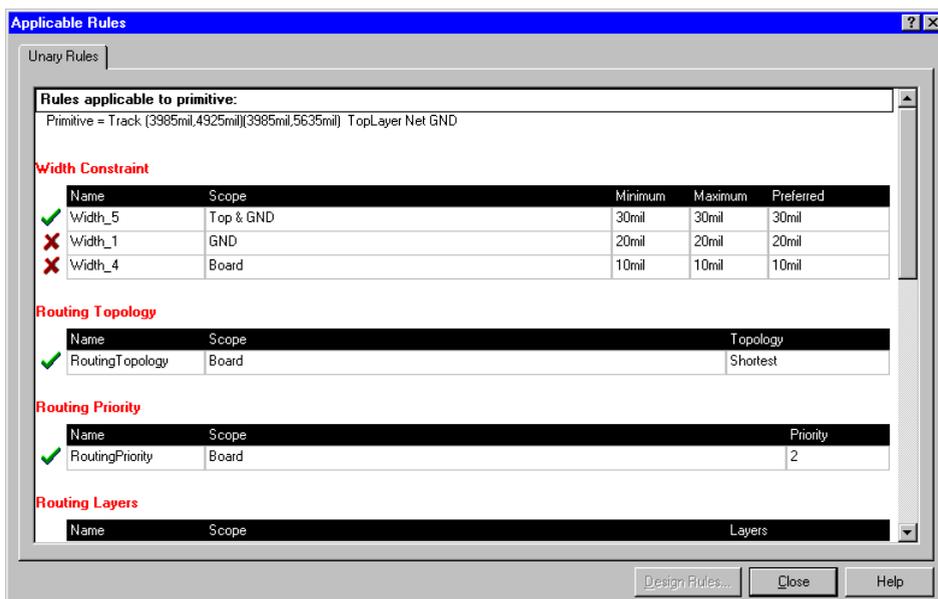


Рис. 6.40. Все правила проектирования, приложенные к указанному объекту, отображаются в диалоговом окне Applicable Rules.

Если необходимо выяснить, почему именно тот или иной объект попал в список нарушений правил проектирования, следует выполнить на нем щелчок правой кнопкой мыши и выбрать **Violations** из всплывающего меню. Появится диалоговое окно **Violation Inspector**, в котором будет подробно описано, что вызвало нарушения, и каких именно правил проектирования.

Классы объектов

Классы представляют собой наборы объектов по какому-либо признаку, предназначенные для выполнения с ними одинаковых операций. Например, все микросхемы памяти на плате могут быть выделены в один класс, а все элементы блока питания — в другой. В этом случае микросхемам памяти можно будет назначить одни правила проектирования, а компонентам блока питания — другие. Использование классов

значительно облегчает идентификацию наборов объектов, особенно при назначении правил проектирования.

Система Protel допускает использование четырех классов объектов: классы цепей, классы компонентов, классы маршрутов и классы контактных площадок. Причем классы компонентов и цепей могут создаваться прямо из панели управления редактором печатных плат.

Классы не являются взаимоисключающими множествами, то есть один объект может принадлежать одновременно к нескольким классам. Это позволяет помещать объекты в несколько классов, которые используются для разных целей. Например, некоторый компонент может входить в класс, определяющий его к компонентам, которые должны быть размещены в указанной области платы (с помощью правила **Room Definition**). Этот же компонент может входить во второй класс, в котором производится группировка по топологическому посадочному месту, и к которому применяется правило вращения **Rotation Rule**; а также и в третий класс, содержащий компоненты, имеющие специфические установки для правила проектирования **Solder Mask Expansion**.

Создание классов

Все четыре типа классов создаются одинаково в диалоговом окне **Object Classes** (рис. 6.41), вызываемом командой меню **Design » Classes**. Далее следует щелкнуть левой кнопкой мыши на вкладке с названием типа создаваемого класса и нажать кнопку **Add** в нижней части диалогового окна. В появившемся после этого окне **Edit Class** следует выбрать нужные объекты из списка **Non-Members** и с помощью кнопок со стрелками перенести их в список **Members**. Отметим, что здесь возможно множественное выделение стандартными для системы Windows методами (клавиши **SHIFT** и **CTRL**).

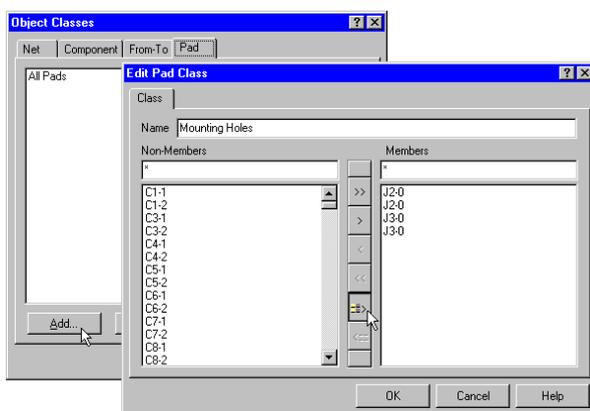


Рис. 6.41. Создание класса контактных площадок из набора выделенных в настоящий момент

Классы компонентов

Класс компонентов может быть создан с помощью специально предназначенного для этих целей модуля — **Component Class Generator**, который позволяет определить набор компонентов, основанный на общих атрибутах компонентов.

Для вызова диалогового окна **Component Class Generator** необходимо во вкладке **Components** диалогового окна **Object Classes** нажать кнопку **Add**. Поле фильтров, находящееся в левой части окна, используется для определения компонентов посредством их атрибутов (рис. 6.42). После выбора фильтра необходимо нажать кнопку **>** для добавления всех компонентов, которые имеют такое значение атрибута. Далее в верхней части окна нужно указать имя класса (**Class Name**) и затем нажать кнопку **OK**.

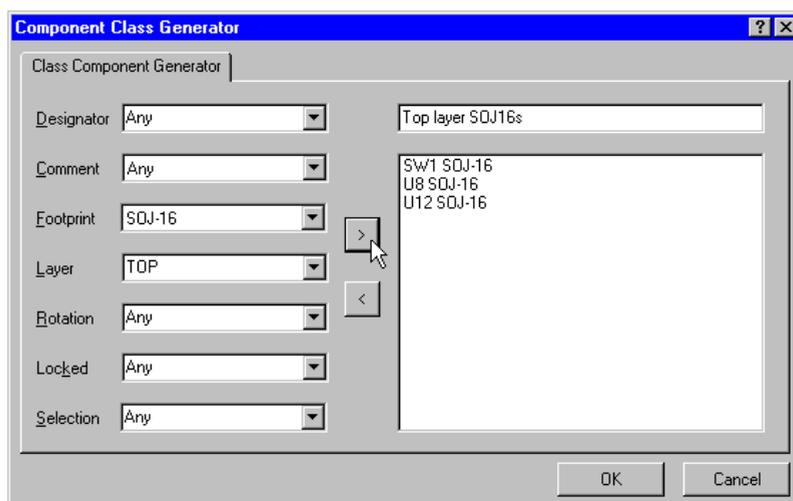


Рис. 6.42. Для быстрого создания классов компонентов по их параметрам используется специальный генератор классов.

Правила проектирования, учитываемые при трассировке (Routing)

Clearance Constraint (размер зазора)

Определение

Определяет минимально допустимый зазор между любыми двумя медными объектами на сигнальном слое. Это правило используется для задания расстояния между проводниками на плате.

Проверка соединений

Обычно эта опция устанавливается в значение **Different Nets** (разные цепи). Установка этой опции в значение **Any Net** (все цепи) может быть применена, когда нужно тестировать переходные отверстия, расположенные слишком близко к контактным площадкам или другим переходным отверстиям, принадлежащим этой же цепи или любой другой цепи.

Как разрешается конфликт дублированных правил

При дублировании правил наивысший приоритет имеет то правило, где назначается самый большой зазор.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах, а также во время автотрассировки.

Routing Corners Rule (углы изгиба проводников)

Определение

Определяет стиль излома проводников, используемый при автотрассировке. Изломы могут быть выполненными под углом 45° или скругленными. Устанавливаемые значения определяют минимальное и максимальное размеры скоса или дуги.

Как разрешается конфликт дублированных правил

При дублировании правила наивысший приоритет имеет опция скругленных углов, затем $90^\circ/45^\circ$ и 90° .

Приложения, учитывающие данные правила

Экспорт в программу трассировки SPECCTRA.

Routing Layers Rule (слои трассировки)

Определение

Определяет слои, которые будут использоваться при автотрассировке.

Как разрешается конфликт дублированных правил

Максимальный приоритет имеет правило с минимальным количеством слоев.

Приложения, учитывающие данные правила

Проверяются во время автотрассировки.

Routing Priority Rule (приоритет трассировки)

Определение

Данное правило присваивает цепи приоритет трассировки. Самый высокий приоритет равен 100, самый низкий — 0. Приоритет трассировки представляет собой относительное значение, используемое для задания порядка автотрассировки цепей.

Как разрешается конфликт дублированных правил

При дублировании выполняются правила с наивысшим приоритетом.

Приложения, учитывающие данные правила

Проверяются во время автотрассировки.

Routing Topology Rule (топология трассировки)

Определение

Данное правило определяет порядок или образец соединения выводов проводниками. По умолчанию редактор печатных плат располагает соединения между выводами таким образом, чтобы полная длина всех соединений была минимальной. Специфические требования могут накладываться на отдельные цепи по нескольким причинам: для высокоскоростных схем, где отражения сигналов должны быть минимизированы, применяется последовательная топология или "цепочка"; для цепей заземления может быть использована топология типа "звезда", что гарантирует наличие общей точки для всех проводников. В описываемом правиле могут быть применены следующие виды топологий:

Shortest (минимальная длина)

Эта топология соединяет все узлы так, чтобы полная длина всех соединений была минимальной..

Horizontal (горизонтальная)

Эта топология соединяет все узлы таким образом, что длина горизонтальных связей относится к длине вертикальных, как 5:1. Этот метод используется для выполнения трассировки в горизонтальном направлении.

Vertical (вертикальная)

Эта топология соединяет все узлы таким образом, что длина вертикальных связей относится к длине горизонтальных, как 5:1. Этот метод используется для выполнения трассировки в вертикальном направлении.

Daisy-Simple (простая цепочка)

Эта топология соединяет все узлы в цепочку один за другим. Порядок следования узлов в цепочке рассчитывается так, чтобы минимизировать общую длину. Если заданы начальная и конечная контактные площадки (**source** и **terminator**), то остальные располагаются по принципу минимизации общей длины (рис. 6.43). Для назначения начальных или конечных контактных площадок необходимо выполнить их редактирование. Если задано несколько начальных (или конечных) контактных площадок, то они соединяются вместе размещаются на конце цепочки.

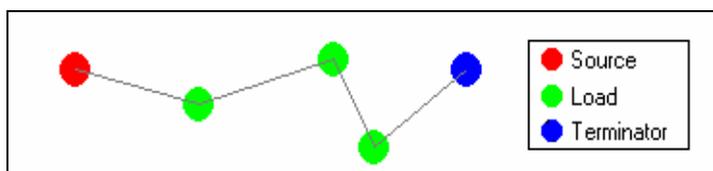


Рис. 6.43. Соединение узлов простой цепочкой Daisy-Simple.

Daisy-Mid Driven (цепочка с началом в центре)

При этой топологии начальный узел (или узлы) размещается в центре цепочки, остальные узлы делятся на две группы и размещаются по разные стороны от начального узла (рис. 6.44). При этом необходимо задать два конечных узла, которые помещаются на концах цепочки. При назначении нескольких начальных узлов они соединяются вместе и размещаются в центре цепочки. Если не определено точно два конечных узла, то используется топология простой цепочки **Daisy-Simple**.

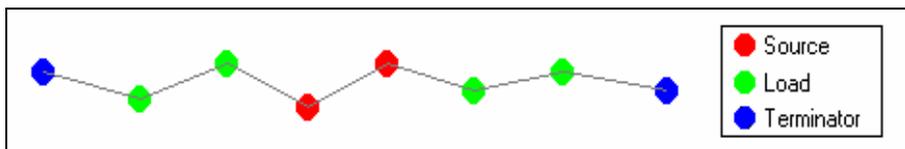


Рис. 6.44. Соединение узлов в цепочку с началом в центре.

Daisy-Balanced (сбалансированные цепочки)

При такой топологии все множество узлов делится на одинаковые по числу узлов цепочки, общее количество которых равно числу конечных узлов (рис. 6.45). Затем эти цепочки соединяются с начальным узлом по схеме "звезда". Несколько начальных узлов соединяются вместе.

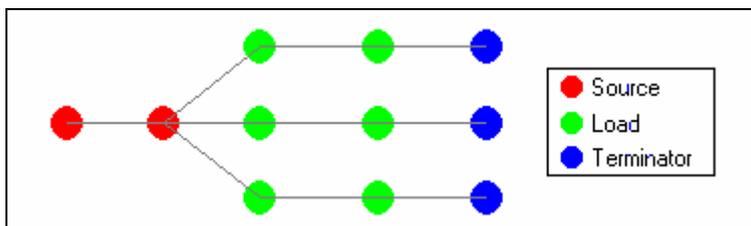


Рис. 6.45. Соединение узлов в сбалансированную цепочку.

Star ("звезда")

При топологии такого типа каждый узел подсоединяется непосредственно к начальному узлу. Если заданы конечные узлы, они подсоединяются после каждого узла. При наличии нескольких начальных узлов они соединяются вместе, как в топологии типа **Daisy-Balanced**.

Как разрешается конфликт дублированных правил

Правила выполняются в следующем порядке: **Star**, **Daisy-Balanced**, **Daisy-Mid Driven**, **Daisy-Simple**, **Horizontal**, **Vertical**, **Shortest**.

Приложения, учитывающие данные правила

Проверяются во время автотрассировки.

Routing Via Style Rule (стиль переходных отверстий)

Определение

Данное правило определяет диаметр пятна металлизации и диаметр круглого переходного отверстия.

Как разрешается конфликт дублированных правил

Приоритет имеют правила с наибольшим размером переходных отверстий.

Приложения, учитывающие данные правила

Задаваемые правилом параметры используются при ручной трассировке при нажатии клавиши * (переход на другой сигнальный слой) или / (соединение с внутренним слоем питания или заземления). Данные параметры могут быть изменены во время размещения отверстия посредством нажатия клавиши **TAB** или в любое другое время.

Программа автотрассировки использует данное правило по умолчанию для всей платы. При пакетной проверке правил проектирования контролируются только атрибуты maximum и minimum.

SMD Neck-Down Constraint (скачок ширины проводника)

Определение

Данное правило определяет максимальное отношение ширины проводника к ширине контактной площадки компонента для поверхностного монтажа, выраженное в процентах.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило в списке.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

SMD to Corner Constraint (минимальное расстояние до изгиба)

Определение

Данное правило определяет минимальное расстояние от центра контактной площадки компонента для поверхностного монтажа до ближайшего изгиба подключенного к ней проводника.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило в списке.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

SMD to Plane Constraint (расстояние до переходного отверстия)

Определение

Данное правило определяет максимальное расстояние от центра контактной площадки компонента для поверхностного монтажа до ближайшего переходного отверстия на внутренний слой питания или заземления.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило в списке.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Routing Width Constraint (ограничение ширины проводников)

Определение

Данное правило определяет минимальную, максимальную и рекомендуемую ширину проводников и дуг на медном слое.

Как разрешается конфликт дублированных правил

Приоритет имеют правила с минимальной разницей минимального и максимального значений.

Приложения, учитывающие данные правила

Рекомендуемое значение ширины проводника используется программой автотрассировки. Минимальное и максимальное значения контролируются программой проверки (DRC) в интерактивном и пакетном режимах.

Правила проектирования, учитываемые при производстве (Manufacturing)

Acute Angle Constraint (ограничение на размер острых углов)

Определение

Данное правило определяет минимально допустимый угол излома проводников. Наличие острых углов может стать проблемой при производстве платы, так как при травлении в точке излома может возникнуть разрыв.

Как разрешается конфликт дублированных правил

Выполняются правила с максимально заданными размерами углов.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Hole Size Constraint (диаметр отверстий)

Определение

Данное правило определяет минимально и максимально допустимые значения диаметра присутствующих на плате отверстий. Может быть задано как абсолютное значение диаметра, так и его относительное значение от размера контактной площадки или кольца переходного отверстия.

Как разрешается конфликт дублированных правил

Приоритет имеет правило с наименьшим минимальным значением и наименьшей разностью максимально и минимального значений.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Layer Pairs (пары слоев)

Определение

Данное правило проверяет соответствие используемых пар слоев парам слоев для сверления, которые определяются из присутствующих на плате контактных площадок и переходных отверстий. Для каждой пары слоев один выбирается **Start Layer**, другой — **End Layer**.

Как разрешается конфликт дублированных правил

Выполняется первое правило с принудительно назначенными парами слоев.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Minimum Annular Ring (минимальный размер контактной площадки)

Определение

Данное правило определяет минимально допустимый размер кольца контактной площадки, который измеряется радиально от края отверстия контактной площадки до ее кромки.

Как разрешается конфликт дублированных правил

Выполняется правило с наибольшим установленным размером кольца.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Paste-Mask Expansion Rule (размер окна в трафарете для нанесения паяльной пасты)

Определение

В данном правиле задается величина, на которую расширяется или сжимается рисунок контактной площадки на слое **Paste Mask**, из которого формируются окна в трафарете для нанесения паяльной пасты. Сжатие рисунка осуществляется при задании отрицательного значения.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наименьшее расширение.

Приложения, учитывающие данные правила

Учитывается при генерации выходных файлов для производства.

Polygon Connect Style (стиль соединения выводов с полигоном)

Определение

Данное правило определяет стиль соединения выводов компонента с металлизированным полигоном. Система допускает три типа соединения: непосредственное (сплошное) соединение, соединение с тепловым барьером и отсутствие соединения.

В случае выбора соединения с тепловым барьером необходимо задать количество и ширину проводников, а также угол их расположения.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, задающее непосредственное соединение.

Приложения, учитывающие данные правила

Используется во время заливки полигонов.

Power Plane Clearance (зазоры на слоях питания)

Определение

Данное правило определяет радиальный зазор, создаваемый вокруг переходных отверстий и контактных площадок, которые проходят сквозь слои питания, но не соединяются с ними.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наибольший зазор.

Приложения, учитывающие данные правила

Учитывается при генерации выходных файлов для производства.

Power Plane Connect Style (стиль соединения выводов со слоем питания)

Определение

Данное правило определяет стиль соединения выводов компонента со слоем питания. Система допускает три типа соединения: непосредственное (сплошное) соединение, соединение с тепловым барьером и отсутствие соединения.

В случае выбора соединения с тепловым барьером необходимо задать количество и ширину проводников, а также угол их расположения, расстояние от центра вывода до зазора и его ширину. Заметим, что слои питания отображаются в негативе, а значит нарисованный на нем примитив будет вытравлен на слое меди.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет непосредственные соединения.

Приложения, учитывающие данные правила

Учитывается при генерации выходных файлов для производства.

Solder-Mask Expansion Rule (размер окна в трафарете для пайки волной)

Определение

В данном правиле задается величина, на которую расширяется или сжимается рисунок контактной площадки на слое **Solder Mask**, из которого формируются окна в трафарете для пайки волной. Чтобы не делать окно в трафарете для какого-либо переходного отверстия, необходимо установить отрицательное значение, превышающее его радиус. Если такие отверстия имеют разные радиусы, то надо задать наибольшее из них.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наибольшее значение.

Приложения, учитывающие данные правила

Учитывается при генерации выходных файлов для производства.

Testpoint Style (стиль контрольных точек)

Определение

В данном правиле задаются допустимые физические размеры контактных площадок и переходных отверстий, используемых в качестве контрольных точек.

Функция **Find Testpoint** и программа автотрассировки используют параметр **Allowed Side** в следующем порядке (по убывания приоритета).

- На нижнем слое (при поверхностном монтаже)
- На верхнем слое (при поверхностном монтаже)
- Переходные отверстия на нижнем слое
- Переходные отверстия на верхнем слое
- Контактные площадки на нижнем слое со сквозными отверстиями

- Контактные площадки на верхнем слое со сквозными отверстиями

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило из списка.

Приложения, учитывающие данные правила

Данное правило проектирования используется функцией **Find Testpoint**, программой автотрассировки, а также проверяется программой проверки (DRC) в интерактивном и пакетном режимах. Значение параметров **Preferred Size** и **Preferred Hole Size** используется программой автотрассировки при добавлении контрольных точек.

Testpoint Usage (использование контрольных точек)

Определение

Данное правило предназначено для определения цепей, требующих добавления контрольных точек. Для идентификации цепей, нарушающих это правило, используется отчет программы проверки (DRC). Модуль **CAM Manager** используется для определения местоположения этих контрольных точек.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило из списка.

Приложения, учитывающие данные правила

Данное правило проектирования используется функцией **Find Testpoint**, программой автотрассировки, а также проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Правила проектирования, задаваемые для высокоскоростных схем (High Speed)

Daisy Chain Stub Length (ограничение на длину шлейфа)

Определение

Данное правило определяет максимально допустимую длину шлейфа для цепей с топологией в виде цепочки.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наименьшую длину шлейфа.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Length Constraint (ограничение длины проводника)

Определение

Данное правило определяет минимальную и максимальную длину проводника.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наименьшую разницу между установленными значениями.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Matched Net Lengths (длины согласованных цепей)

Определение

Данное правило определяет меру разницы длин цепей. Редактор печатных плат определяет самую длинную цепь (в указанной области) и сравнивает ее с другими цепями в этой области.

Диалоговое окно этого правила также позволяет определить, каким образом нужно изменить длину тех цепей, которые не удовлетворяют требованиям согласований длин цепей, после чего редактор печатных плат автоматически добавляет дополнительные сегменты к цепям, длины которых нужно выровнять.

Если возникает необходимость в том, чтобы редактор печатных плат сделал попытку согласовать длины цепей с помощью добавления дополнительных сегментов, нужно установить соответствующие настройки в диалоговом окне правила проектирования **Matched Length Rules**, а затем выбрать команду меню **Tools » Equalize Nets**. Правило согласования длин будет применено к цепям, установленным посредством этого правила, и затем к тем цепям, которые не удовлетворяют правилу, будут добавлены дополнительные сегменты. Вероятность успешного применения данного правила зависит от количества свободного пространства, доступного для добавления дополнительных сегментов, и от стиля, используемого для их добавления. Наиболее компактным считается стиль добавления сегментов под прямым углом (**90 degree**), а наименее компактным — стиль с округлением углов (**Rounded**).

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наименьшее значение параметра *Tolerance*.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Maximum Via Count (максимальное число переходных отверстий)

Определение

Данное правило определяет максимально допустимое количество переходных отверстий.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое определяет наименьшее количество переходных отверстий.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Parallel Segment Constraint (ограничение на длину параллельных сегментов)

Определение

Данное правило определяет длину параллельных сегментов двух проводников в зависимости от заданного расстояния между этими сегментами. Заметим, что данное правило тестирует только два сегмента проводников и не тестирует наборы сегментов. Для оценки уровня перекрестных искажений, который является функцией от длины и размера зазора, необходимо использовать несколько таких правил для множества параллельно проложенных сегментов цепи.

Как разрешается конфликт дублированных правил

Установка данного правила не создает конфликта дублированных правил.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Via Under SMD Constraint (переходные отверстия под SMD элементами)

Определение

Данное правило устанавливает возможность размещения переходных отверстий во время автоматической трассировки под контактными площадками для устройств, использующих технологию поверхностного монтажа.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое не разрешает размещение переходных отверстий.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Правила проектирования, используемые при размещении компонентов (Placement)

Component Clearance Constraint (расстояние между компонентами)

Определение

Данное правило устанавливает минимально допустимое расстояние между компонентами.

В списке **Check Mode** задается один из трех режимов проверки:

Quick Check — быстрая проверка — для определения формы компонента использует ограничивающий его прямоугольник, то есть минимально возможный прямоугольник, который охватывает все примитивы, входящие в данный компонент.

Multi Layer Check — многослойная проверка — также использует ограничивающий прямоугольник, но рассматривает контактные площадки со сквозными отверстиями с таким расчетом, что допускается расположение компонентов, использующих технологию поверхностного монтажа, друг под другом, но на разных сторонах платы.

Full Check — полная проверка — использует точную форму компонента, которая включает все примитивы, его составляющие. Рекомендуется применять в случаях, когда проект содержит большое количество компонентов круглой или неправильной формы.

Как разрешается конфликт дублированных правил

Приоритет имеет правило с наименьшим зазором.

Приложения, учитывающие данные правила

Установленные правила проверяются в режимах интерактивной, пакетной проверки правил проектирования, а также при автоматическом размещении компонентов программой Cluster Placer.

Component Orientation Rule (ориентация компонентов)

Определение

Данное правило определяет допустимую ориентацию компонентов. Разрешается назначать одновременно несколько видов ориентации, что позволяет программе автоматического размещения выбирать любую из них.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое устанавливает наименьший угол поворота.

Приложения, учитывающие данные правила

Учитывается во время автоматического размещения компонентов с помощью программы Cluster Placer.

Nets to Ignore (игнорирование цепей)

Определение

Данное правило определяет, какие цепи должны быть проигнорированы при выполнении процедурой Cluster Placer операции автоматического размещения. Игнорирование цепей питания может оказать содействие в быстром и качественном размещении компонентов. Если проект имеет большое количество двухвыводных компонентов, которые соединяются с некоторой цепью питания, то игнорирование этой цепи приведет к тому, что компоненты будут сгруппированы не на основе указанной цепи питания, а на основе другой присоединенной к компоненту цепи.

Как разрешается конфликт дублированных правил

Конфликты исключены.

Приложения, учитывающие данные правила

Учитывается во время автоматического размещения компонентов с помощью программы Cluster Placer.

Permitted Layers Rule (разрешенные слои)

Определение

Данное правило определяет, на каких слоях могут быть размещены компоненты программой Cluster Placer. Программа Cluster Placer не может изменить слой, где будут размещаться компоненты, поэтому нужно устанавливать его до запуска программы размещения.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое устанавливает наименьшее количество слоев.

Приложения, учитывающие данные правила

Учитывается во время автоматического размещения компонентов с помощью программы Cluster Placer.

Room Definition (области размещения)

Определение

Данное правило определяет прямоугольную область **Room**, в которой либо разрешено, либо запрещено размещать некоторый набор объектов. Области размещения определяются с помощью команды меню **Place » Room**. Редактирование областей производится аналогично любым другим объектам на чертеже.

Перемещение компонентов в области размещения производится с помощью кнопки **Arrange Components within Room**, расположенной на панели инструментов **Component Placement**. При включенном правиле **Room Definition** при переносе одного из компонентов класса в такую область, все компоненты этого класса также будут перенесены сюда.

Как разрешается конфликт дублированных правил

Выполняются все назначенные правила.

Приложения, учитывающие данные правила

Установленные правила проверяются в режимах интерактивной, пакетной проверки правил проектирования, а также при автоматическом размещении компонентов программой Cluster Placer.

Правила проектирования, учитывающие анализ целостности сигналов (Signal Integrity)

Для того, чтобы использовать правила, учитывающие анализ целостности сигналов, необходимо установить правила **Signal Stimulus Rule** и **Supply Nets rules**. Кроме того, на вкладке **Signal Integrity** диалогового окна **Preferences** должна быть сконфигурирована таблица **Designator Mapping**.

Impedance Constraint (требования к импедансам)

Определение

Данное правило определяет минимально и максимально допустимый импеданс цепи. Импеданс цепи является функцией геометрии проводника и его проводимости, материалов окружающих диэлектриков (основной материал платы, межслойный изоляционный материал, покрытие паяльной маски и др.) и физической геометрии платы (расстояние до других проводников по оси Z).

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое устанавливает наименьшую разницу значений импедансов.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Overshoot — Falling Edge (отрицательный выброс на заднем фронте импульса)

Определение

Данное правило определяет максимально допустимый отрицательный выброс (затухающие колебания относительно низкого значения напряжения) на заднем фронте импульса сигнала.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьший выброс.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Overshoot — Rising Edge (положительный выброс на переднем фронте импульса)

Определение

Данное правило определяет максимально допустимый положительный выброс (затухающие колебания относительно высокого значения напряжения) на переднем фронте сигнала.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьший выброс.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Signal Base Value (низкое значение уровня сигнала)

Определение

Определяет максимально допустимое значение напряжения сигнала низкого уровня.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьшее допустимое значение.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Signal Flight Time Falling Edge (задержка заднего фронта)

Определение

Flight Time — это время задержки сигнала, обусловленное структурой внутренних соединений схемы. Оно вычисляется как время, которое затрачивается на то, чтобы напряжение на реальном входе достигло порога, минус время, которое затрачивается на то, чтобы напряжение на опорной нагрузке (подсоединенной непосредственно к выходу) достигло порога.

Данное правило определяет максимально допустимое время задержки заднего фронта сигнала.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьшее значение времени задержки.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Signal Flight Time Rising Edge (задержка переднего фронта)

Определение

Flight Time — это время задержки сигнала, обусловленное структурой внутренних соединений схемы. Оно вычисляется, как время, которое затрачивается на то, чтобы напряжение на реальном входе достигло порога, минус время, которое затрачивается на то, чтобы напряжение на опорной нагрузке (подсоединенной непосредственно к выходу) достигло порога.

Данное правило определяет максимально допустимое время задержки переднего фронта сигнала.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьшее значение времени задержки.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Signal Stimulus (входные сигналы)

Определение

Данное правило определяет характеристики входных сигналов, которые используются при анализе целостности сигналов. Это сигналы, которые подаются на каждый входной вывод тестируемой цепи и формируются на выходных выводах. Во время проверки правил проектирования возвращается наилучший возможный результат.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило из списка.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Signal Top Value (напряжение сигнала высокого уровня)

Определение

Данное правило определяет минимально допустимое значение напряжения сигнала высокого уровня.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наибольшее допустимое значение.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Slope — Falling Edge (крутизна заднего фронта импульса)

Определение

Данное правило определяет максимально допустимое значение крутизны заднего фронта импульса. Под этим значением здесь подразумевается время, затраченное на изменение уровня сигнала от порогового напряжения (**VT**) до действительного значения напряжения низкого уровня (**VIL**).

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьшее допустимое значение.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Slope — Rising Edge (крутизна переднего фронта импульса)

Определение

Данное правило определяет максимально допустимое значение крутизны переднего фронта импульса. Под этим значением здесь подразумевается время, затраченное на изменение уровня сигнала от порогового напряжения (**VT**) до действительного значения напряжения высокого уровня (**VIH**).

Как разрешается конфликт дублированных правил

Приоритет имеет правило, устанавливающее наименьшее допустимое значение.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Undershoot — Falling Edge (положительный выброс на заднем фронте импульса)

Определение

Данное правило определяет максимально допустимое значение положительного выброса (затухающие колебания относительно низкого значения напряжения) на заднем фронте сигнала.

Как разрешается конфликт дублированных правил

Приоритет правило, устанавливающее наименьшее допустимое значение выброса.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Undershoot — Rising Edge (отрицательный выброс на переднем фронте импульса)

Определение

Данное правило определяет максимально допустимое значение отрицательного выброса (затухающие колебания относительно высокого значения напряжения) на переднем фронте сигнала.

Как разрешается конфликт дублированных правил

Приоритет правило, устанавливающее наименьшее допустимое значение выброса.

Приложения, учитывающие данные правила

Учитывается во время анализа целостности сигналов.

Прочие правила проектирования (Other)

Short Circuit Constraint (короткозамкнутые цепи)

Определение

С помощью данного правила проверяется наличие короткого замыкания между примитивами различных цепей. Наличие короткого замыкания констатируется, когда два объекта, принадлежащие цепям с различными именами касаются друг друга.

Как разрешается конфликт дублированных правил

Приоритет имеет правило, которое не разрешает замыкание цепей.

Приложения, учитывающие данные правила

Учитываются при пакетной и интерактивной проверке правил проектирования.

Un-Connected Pin Constraint (неприсоединенные выводы)

Определение

Данное правило служит для выявления выводов, не соединенных с проводниками на плате.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило из списка.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Un-Routed Nets Constraint (неразведенные цепи)

Определение

Данное правило проверяет статус завершения всех цепей, попавших в указанную область. Если трассировка некоторой цепи выполнена не до конца, то каждая незавершенная часть цепи (**sub-net**) заносится в список, где также указывается коэффициент завершения, рассчитываемый как отношение выполненного числа соединений к полному числу в процентах.

Как разрешается конфликт дублированных правил

Приоритет имеет первое правило из списка.

Приложения, учитывающие данные правила

Данное правило проверяется программой проверки (DRC) в интерактивном и пакетном режимах.

Примеры использования правил проектирования

Использование области действия правила Net Class Rule

Рассмотрим случай, когда имеются две группы цепей — сигнальные цепи (класс **Signal**) и цепи с низким напряжением питания (класс **LV Power**). Допустим, необходимо между цепями из разных классов сохранять расстояние в 50 тысячных дюйма, а между цепями, принадлежащими одному классу, — расстояние в 10 тысячных дюйма.

Сначала может показаться, что нужно просто установить одно правило **Clearance Rule** с зазором, равным 10 тысячных дюйма и областью действия **Whole Board to Whole Board** (вся плата — вся плата), и второе правило **Clearance Rule** с зазором, равным 50 тысячных дюйма и областью действия **Net Class to Whole Board** (класс цепи — вся плата). Однако, такая комбинация правил будет поддерживать зазор, равный 50 тысячных дюйма, между всеми цепями класса **LV Power**.

Для преодоления этой ситуации необходимо создать два класса цепей и установить правило **Clearance Rule** с зазором, равным 50 тысячных дюйма, с областью действия **Net Class to Net Class** (класс цепи — класс цепи). Первое правило, которое определяет зазор в 10 тысячных дюйма, с областью действия вся плата — вся плата создаст нужный зазор между цепями внутри обоих классов, тогда как второе правило с областью действия класс цепи — класс цепи создаст зазор между цепями из разных классов (рис 6.46).

Использование правил Mask Expansions для координатных меток

Координатные метки представляют собой специальные элементы на топологии, используемые для оптического центрирования печатной платы при автоматической сборке. Очень важным фактом является обеспечение того, чтобы маски нанесения пасты и пайки соответствовали требованиям на координатные метки, принятые на заводе-изготовителе, где будут собираться данные платы.

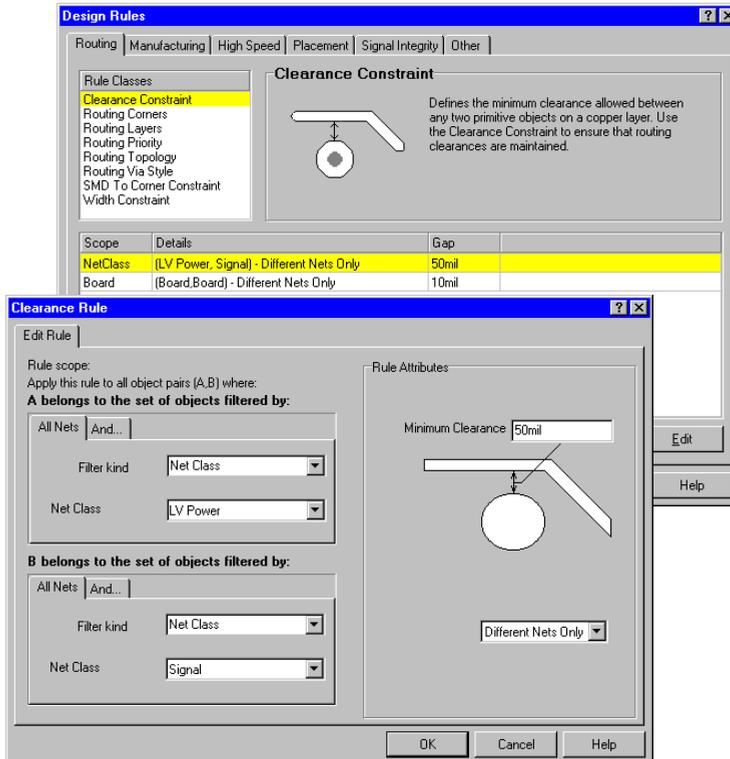


Рис. 6.46. Пример задания зазора между цепями различных классов.

Для использования координатных меток в проекте необходимо:

1. Создать координатную метку, как компонент в библиотеке. Для этого проще всего использовать однослойную контактную площадку и присвоить ей обозначение **FID**. Далее нужно сохранить созданный компонент, назначив ему имя по вашему желанию.
2. Разместить созданные координатные метки согласно технологическим требованиям.
3. Создать класс компонентов **Fiducials**, который бы включал все координатные компоненты.
4. Добавить правило **Solder Mask Expansion** в диалоговом окне **Design Rules**. Установить тип фильтра (поле **Filter Kind**) для области действия правила в значение **Component Class** и выбрать значение **Fiducials** в поле **Component Class**. Установить соответствующее значение параметра **Expansion** в секции атрибутов (рис. 6.47).

Эти действия гарантируют, что для всех компонентов в классе **Fiducials** значение **Expansion** будет применено для создания маски для пайки волной. Если для отдельных координатных меток необходимо задать другое значение **Expansion**, тогда необходимо удалить этот компонент из класса **Fiducials** и применить другое правило проектирования к этому элементу.

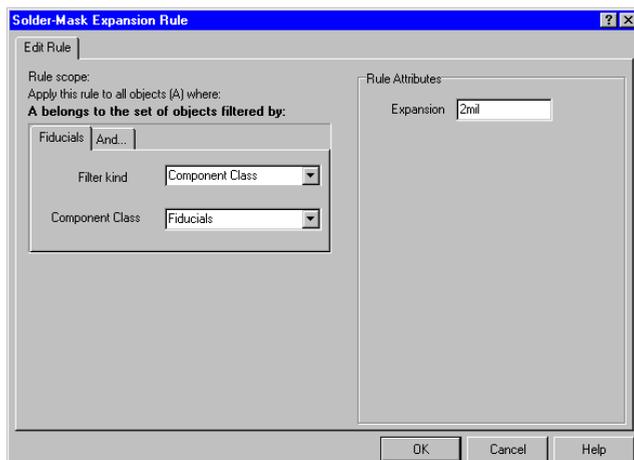


Рис. 6.47. Использование правил Mask Expansions для координатных меток.

Для координатных меток также необходимо определить правило **Paste Mask Expansion**, чтобы гарантировать, что трафарет для них не изготавливается. Для того, чтобы исключить отверстие на максе нанесения пасты необходимо:

5. Добавить правило **Paste Mask Expansion** в диалоговом окне **Design Rules**. Используя ранее описанный подход, установить область действия правила для класса компонентов и выбрать класс **Fiducials**.
6. Установить большое отрицательное число в поле **Expansion**, которое было бы больше радиуса самой большой используемой в проекте координатной метки (рис. 6.48).

Отрицательное число в поле **Expansion** сообщит редактору печатных плат, что нужно радиально сузить отверстие в данной маске на указанную величину. Когда величина сужения больше, чем радиус самой большой координатной метки, тогда отверстия на слое **Paste Mask** исчезают.

Применение правила Clearance к части цепи

Правила проектирования могут быть применены к отдельной части цепи.

1. Необходимо идентифицировать критические части цепи посредством определения маршрутов **From-To** для этих частей цепи.

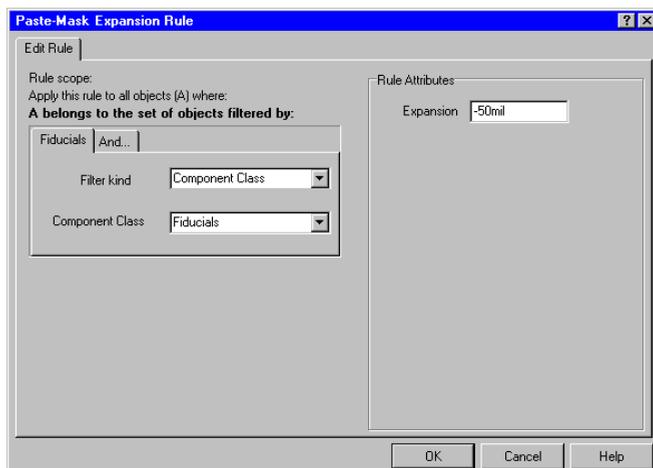


Рис. 6.48. Задание параметра Paste Mask для всех реперных контактных площадок

Для создания маршрута **From-To** нужно воспользоваться командой меню **Design » From-To Editor**. Определение маршрутов необходимо для идентификации критических участков цепи. Дополнительная информация по вопросу определения маршрутов From-To представлена в разделе *Что такое связанность и топология*. Если для идентификации всех критических частей цепи требуется более одного маршрута, рекомендуется создать класс **From-To**. Классы создаются в диалоговом окне **Object Classes**, которое вызывается с помощью команды меню **Design » Classes**. После идентификации критических частей цепи можно выполнять добавление правил проектирования.

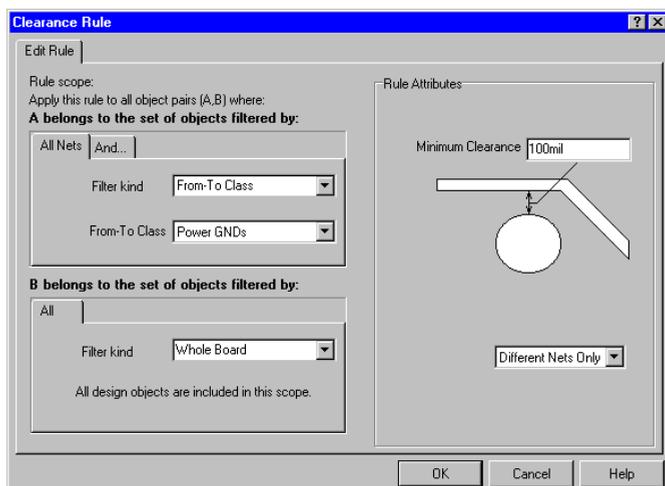


Рис. 6.49. Задание специфического зазора для критического участка цепи

2. В диалоговом окне **Design Rules** добавьте правило **Clearance Constraint**.
3. Установите тип фильтра для области действия в значение **From-To** или **From-To Class**, если был создан класс соответствующий класс (рис. 6.49).
4. Установите необходимое значение зазора (**Clearance**). Оно определяет минимально допустимое расстояние между любым объектом, принадлежащим указанной части цепи, и любым другим объектом на плате.

Предотвращение соединения переходных отверстий с внутренними слоями питания и заземления.

В системе Protel 99 SE имеется возможность соединения переходных отверстий с внутренним слоем питания или заземления. Если имя цепи при размещении переходного отверстия совпадает с именем такого слоя, то оно будет автоматически соединено с ним. Стиль соединения определяется правилом проектирования Power Plane Connect Style, назначенным для данной цепи.

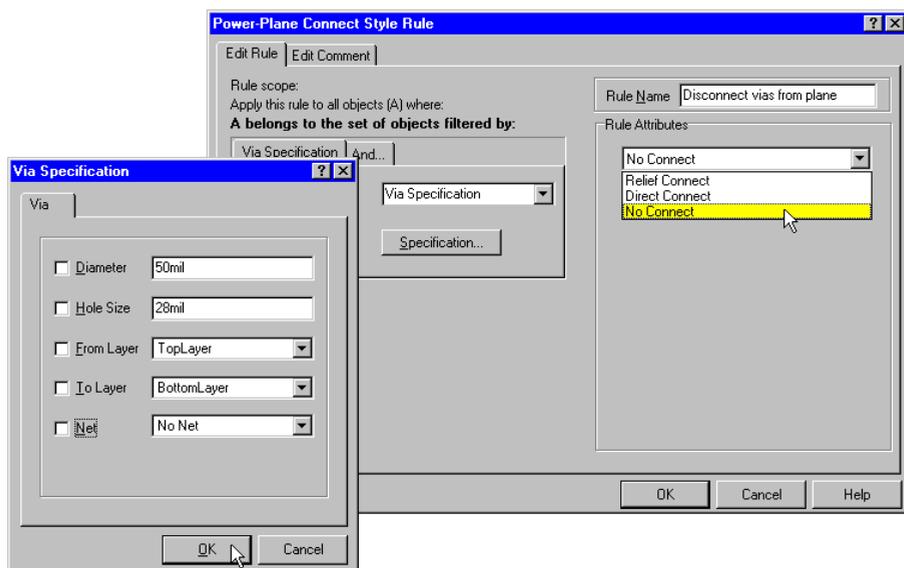


Рис. 6.50. Выключите все опции идентификации в окне Via Specification для применения правила ко всем без исключения переходным отверстиям.

Предотвращение соединения также выполняется с помощью этого правила. Для этого необходимо установить область действия в значение **Via Specification** и в диалоговом окне **Via Specification** выключить все атрибуты, чтобы правило применялось ко всем переходным отверстиям. Далее следует установить значение стиля соединения в **No Connect**, что означает отсутствие соединения со слоем (рис. 6.50).

Задание разных зазоров между объектами на верхнем слое

Предположим, что требуется выдержать расстояние в 6 тысячных дюйма между проводниками на верхнем слое, а зазор между проводником и контактной площадкой или переходным отверстием должен быть 8 тысячных дюйма. Первое условие может быть выполнено с помощью задания соответствующего правила, примененного ко всей плате. Второе специфическое требование реализуется посредством введения правила, задание которого показано на рисунке 6.51.

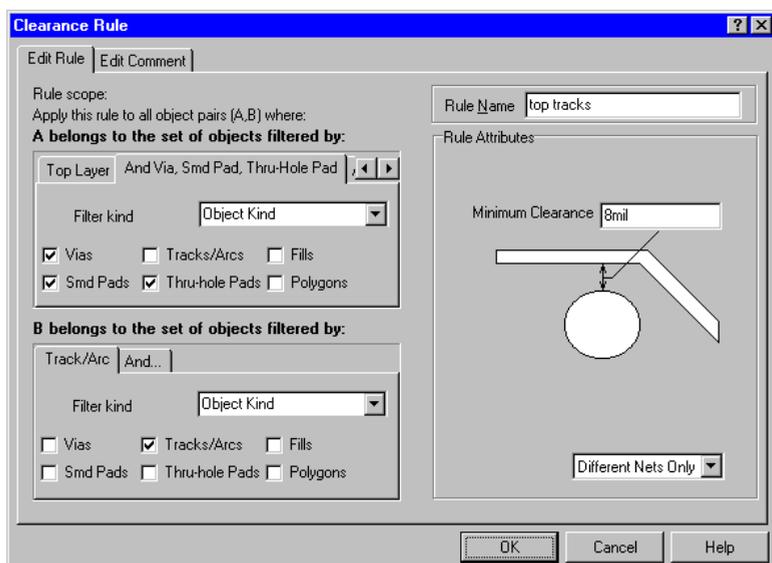


Рис. 6.51. Использование сложной области действия дает возможность установить больший зазор между проводниками и контактными площадками или переходными отверстиями, чем просто между проводниками.

Техника размещения компонентов и инструменты для этих целей

После определения требований на границы платы и keep-out области, а также успешной загрузки списка соединений все готово для выполнения операции размещения компонентов. При первой загрузке списка соединений компоненты выравниваются построчно к правому контуру платы (если он существует) или к центру рабочего пространства (50000, 50000). Затем компоненты могут быть выровнены вручную, с помощью интерактивных инструментов размещения, посредством инструментов автоматического размещения или с применением комбинации перечисленных методов.

В настоящем справочнике термин размещение компонентов означает процесс расположения или позиционирования компонентов на чертеже печатной платы.

◆ Удачное размещение компонентов является фундаментальной основой процесса проектирования. Степень сложности производства и трассировки платы во многом определяется качеством размещения компонентов.

Основные опции размещения

Существует несколько опций, которые управляют поведением редактора печатных плат в процессе ручного размещения компонентов.

Snap to Center на вкладке **Options** (диалоговое окно **Preferences**)

При включении данной опции при перемещении компонент удерживается за точку привязки, в противном случае — за то место, где был выполнен щелчок мыши. Если точка привязки во время выбора компонента находится за пределами экрана, указатель мыши при включенной опции перепрыгивает в точку привязки. При выключенной опции компонент подтягивается к указателю мыши.

Protect Locked Objects на вкладке **Options** (диалоговое окно **Preferences**)

При включении данной опции запрещается перемещение объектов с включенным атрибутом **Locked**, однако, эта опция игнорируется при одновременном перемещении нескольких компонентов. При выключении данной опции при перемещении заблокированных объектов будет выдаваться запрос на подтверждение выполняемого действия.

Rotation Step на вкладке **Options** (диалоговое окно **Preferences**)

Здесь указывается значение, на которое будет повернут компонент (или несколько выделенных компонентов), при нажатии клавиши **SPACEBAR** во время перемещения.

Draft Thresholds на вкладке **Display** (диалоговое окно **Preferences**)

Строки текста, с высотой строки равной или большей указанного в окне **Strings** значения, будут видны как текст. Если же высота строки составляет всего несколько пикселей, то текст будет выводиться в виде прямоугольника.

Component X and Y Grids на вкладке Options (диалоговое окно Document Options)

Здесь задается шаг сетки размещения компонентов по оси X и Y.

Блокирование компонентов

Некоторые компоненты, например, краевые разъемы, могут быть зафиксированы на своем месте. Для этого необходимо выполнить двойной щелчок мыши по компоненту и в появившемся диалоговом окне Component на вкладке **Properties** установить флажок **Locked**.

Расстояние между компонентами

Минимальное расстояние между компонентами задается правилом проектирования **Component Clearance Constraint**, расположенным на вкладке **Placement** диалогового окна **Design Rules**. Это правило учитывается как при интерактивной, так и при пакетной проверке (DRC). Включение данного правила в проверку осуществляется в диалоговом окне **Design Rules Check**, вызываемом командой меню **Tools » Design Rules Check**.

Перемещение компонентов

Для ручного размещения компонентов используется команда меню **Edit » Move » Component**. В строке статуса появляется запрос на выбор нужного компонента (**Select Component**). Выбор осуществляется с помощью щелчка мыши, после чего компонент приклеивается к указателю мыши и может быть перемещен в нужное место платы. Также компонент может быть перемещен посредством нажатия левой кнопки мыши и удержания ее во время перемещения.

Поворот и перенос компонентов на другой слой

Поворот компонентов можно выполнять несколькими способами. Компонент может быть повернут, когда он захвачен указателем мыши. В этом случае вращение против часовой стрелки осуществляется с помощью клавиши **SPACEBAR**, по ходу часовой стрелки — комбинации клавиш **SHIFT+SPACEBAR**. Значение угла поворота задается на во вкладке Options диалогового окна Preferences.

Для поворота группы объектов сначала необходимо их выделить, а затем воспользоваться специальной командой меню **Edit » Move » Rotate Selection**. После этого появятся запросы на задание угла вращения и указание центра вращения (**Reference Point**). Текущее выделение также может быть повернуто посредством нажатия клавиши **SPACEBAR** после запуска команды **Move Selection**.

Для переноса компонента на нижнюю сторону платы в тот момент, когда этот компонент захвачен указателем мыши используется клавиша **L**. Чтобы перевернуть ранее размещенный компонент, необходимо выполнить двойной щелчок мыши по этому компоненту и в появившемся окне редактирования во вкладке **Properties** изменить атрибут **Layer** (слой).

Динамическое повторное соединение с отображением линий соединения

Основным критерием размещения того или иного компонента на плате является минимизация длин соединений данного компонента, поэтому для пользователя важно видеть виртуальные линии связи во время перемещения компонента.

Однако, на платах с высокой плотностью размещения компонентов зачастую трудно разобраться в лабиринте соединительных линий. Другой важный факт, который необходимо учитывать, состоит в том, что при перемещении компонента по плате линии соединения могут не отражать точную связанность компонентов. Это происходит, когда линии соединены с другими выводами в той же цепи, расположенными близко к новому месту компонента.

Пакет Protel 99 SE включает функцию динамического повторного соединения с "умным" алгоритмом отображения виртуальных линий связи. В этом случае при перемещении отдельного компонента, группы выделенных компонентов или объединения (union) компонентов все виртуальные линии временно не отображаются, за исключением тех, которые соединяются с перемещаемым компонентом (рис. 6.52). По мере перемещения эти цепи анализируются и повторно соединяются, что облегчает выбор места для данного компонента.

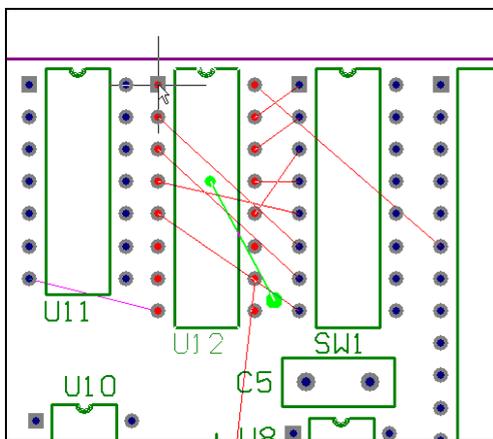


Рис. 6.52. Все линии связи перемещаемого компонента будут отображаться автоматически.

Линии соединения могут быть скрыты с помощью команды меню **View » Connections » Hide All**.

Заметим, что все линии соединения можно временно скрыть при перемещении компонента (включая те, которые отображаются во время работы вышеописанного алгоритма) посредством нажатия на клавишу **N**. То есть, эта клавиша временно отключает анализатор связанности.

В процессе перемещения компонента анализатор длин связей постоянно оценивает качество размещения, основываясь на длинах соединений, и отображает зеленый (хороший) или красный (плохой) вектор, показывающий текущее качество размещения. Дальний конец вектора показывает место для компонента (компонентов), размещение в котором минимизирует общую длину линий связи.

В процессе ручного размещения список соединений автоматически оптимизируется при перемещении компонента на новое место. Если производится перемещение компонента, имеющего большое количество выводов (свыше 500), обновление списка соединений может выполняться слишком медленно. Во избежание этого рекоменду-

ется временно отключить оптимизацию посредством скрытия соединительных линий (клавиша **N**).

Обмен информацией между принципиальной схемой и печатной платой

Передача проекта из редактора схем в редактор плат — это не только процесс передачи информации о компонентах и их связанности. Также существует структурная информация о проекте, воплощенная в принципиальной схеме: в многолистовых проектах отдельные листы разбивают его на логические группы компонентов, расположенных рядом. Такая информация делает проект принципиальной схемы идеальной платформой для управления процессом размещения компонентов на печатной плате. Пакет Protel 99 SE имеет специальные функции, которые помогают передавать такого рода информацию в редактор печатных плат.

Создание классов и областей размещения (Room) компонентов на основе данных из принципиальной схемы

Во многих проектах конфигурация компонентов на листах принципиальной схемы отражает желаемый вариант их размещения на печатной плате. Пакет Protel 99 SE включает функции, помогающие передавать такую информацию о группировке компонентов из редактора принципиальных схем в редактор печатных плат.

Классы компонентов могут быть созданы автоматически из каждого листа схемы во время передачи проектной информации в редактор печатных плат. Для этого необходимо включить опцию **Generate Component Classes and Placement Rooms for all Schematic Sheets in Project** (генерировать классы компонентов и области размещения для всех листов схемы в проекте), расположенную в диалоговом окне **Update Design** (команда меню редактора схем **Update PCB**).

Когда класс компонентов создается для каждого листа схемы, в него включаются все без исключения расположенные на нем компоненты. Каждому классу присваивается имя, составленное из имени соответствующего листа (пробелы исключаются). Если в проекте содержатся многосекционные компоненты, и разные секции одного компонента расположены на разных листах, компонент включается к класс того листа, где находится первая его секция. Для каждого класса компонентов создается своя область размещения (**Room**). Более подробная информация об областях размещения приведена в подразделе *Работа с областями размещения*.

Выбор компонента из редактора схем

Другой мощной функцией, помогающей понять связь между схемой и платой, является возможность выделения компонента на плате из выделения на схеме. Для этого необходимо выделить нужный компонент на схеме и выволнить команду **Tools » Select PCB Components** меню редактора схем. Система автоматически переключится в редактор печатных плат, выделит нужный компонент и изменит масштаб просмотра чертежа платы для наилучшего его отображения. При выделении на схеме нескольких компонентов после выполнения описанной операции все они будут выделены в редакторе печатных плат и из них можно будет создать класс или группу компонен-

тов. Процесс создания класса компонентов из текущего выделения будет описан несколько позднее в пункте *Создание группы компонентов*. Работа с группами компонентов будет описана в пункте *Использование групп компонентов*. Имеется также возможность быстро выбрать компоненты из большой группы и упорядочить в некоторой прямоугольной области с помощью функции **Arrange Within Rectangle**, что будет описано в подразделе *Интерактивное размещение компонентов* раздела *Техника размещения компонентов и инструменты для этих целей*.

Горячая связь между схемой и печатной платой



Перейти от нужного элемента на схеме к нему же на чертеже печатной платы можно с помощью функции горячей связи (**Cross Probe**) двух редакторов. Запуск функции осуществляется нажатием кнопки **Cross Probe**, расположенной на главной панели инструментов редактора схем. Система запросит вас указать элемент на схеме, после чего система автоматически переключится в редактор печатных плат, выделит нужный компонент и изменит масштаб просмотра чертежа платы для наилучшего его отображения. Аналогичным образом можно произвести обратное действие.

Использование групп компонентов

Группами (**union**) компонентов называются специальные наборы компонентов, работать с которыми можно как с блоками. При перемещении группы ее компоненты сохраняют свое относительное положение, что в некоторых случаях чрезвычайно полезно.

Создание группы компонентов



Для создания группы компонентов необходимо выделить компоненты, из которых она будет сформировано, и нажать кнопку **Create Union**, расположенную на панели инструментов **Component Placement**. Заметим, что каждый компонент может принадлежать только одной группе.

Удаление компонентов из группы



Для удаления компонентов из группы необходимо нажать кнопку **Break Union**, расположенную на панели инструментов **Component Placement**, щелкнуть по одному из компонентов группы (то есть указать ее), в появившемся диалоговом окне **Confirm Break Component Union** выбрать компоненты для удаления и нажать кнопку **OK**.

Разбиение групп компонентов

Для разбиения всех существующих групп компонентов используется команда меню **Tools » Convert » Break All Component Unions**.

Работа с областями размещения

Прямоугольный участок платы, выделенный под размещение некоторого набора компонентов называется областью размещения (**Placement Room**). После добавления компонентов область размещения он становится связанным с ней и перемещается вместе с ней.

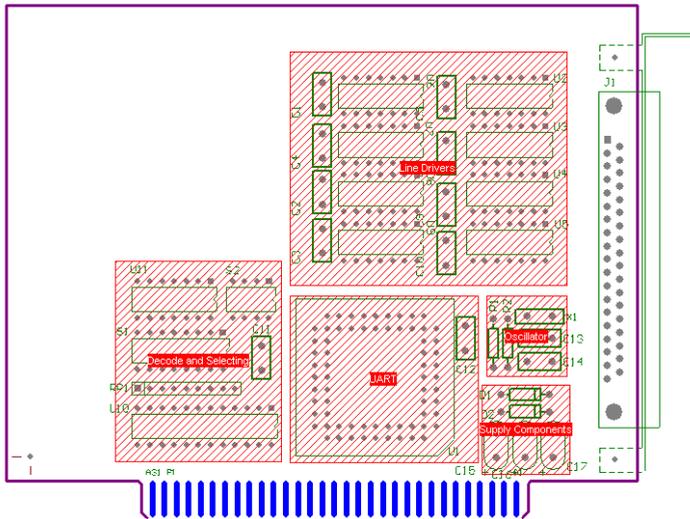


Рис. 6.53. Использование областей размещения (Placement Room) значительно упрощает процесс размещения компонентов

Правило проектирования **Room Definitions** определяет возможность или невозможность размещения данного компонента внутри данной области и проверяется как в интерактивном, так и пакетном режимах. Области размещения также могут учитываться при автоматическом размещении компонентов с помощью программы Cluster Place, причем в этом случае за счет блокирования компонентов можно разбить процесс на этапы и более качественно выполнить размещение.

Создание областей размещения



Области размещения компонентов можно задать с помощью команды меню **Place » Room** или кнопки **Place Room**, расположенной на панели инструментов **PlacementTools**, а также из вкладки **Placement** диалогового окна **Design Rules**. Области размещения можно создавать на нижнем или на верхнем (по умолчанию) слоях. После создания на области выполнения двойной щелчок левой кнопкой мыши и в появившемся окне отредактировать ее параметры.

Привязка компонентов к области размещения

Область действия правила проектирования **Room Definition** определяет набор компонентов, которые могут быть размещены в данной области размещения. Для изменения набора компонентов надо отредактировать область действия этого правила.

Изменение размеров и местоположения области размещения

Размеры области размещения могут быть изменены стандартным для всех других объектов редактора печатных плат методом — выделением фокусом и перетаскиванием появившихся после этого маркеров-манипуляторов.

Заметим, что при перемещении области размещения все привязанные к нему компоненты автоматически перемещаются вместе с ней. Для предотвращения такого перемещения необходимо выключить правило **Room Definition** в диалоговом окне **Design Rules** (снять флажок **Enable**, относящийся к данному правилу). Имеется возможность заблокировать область от случайного перемещения.

Сборка компонентов в пределы области размещения



Пользователю доступна функция быстрой сборки всех привязанных к области размещения компонентов в ее пределы. Запуск этой функции выполняется с помощью кнопки **Arrange Within Room**, расположенной на панели инструментов **ComponentPlacement**.

Интерактивное размещение компонентов

Редактор печатных плат включает интерактивные инструменты размещения компонентов, представленные на панели инструментов **ComponentPlacement** (рис. 6.54).

Инструменты для выравнивания компонентов

Align left edge of selected components (выравнивание по левому краю)

В этом случае для выравнивания используется левый край самого левого компонента из набора выделенных. Компоненты расставляются согласно установленным правилам проектирования, определяющим минимальное расстояние между компонентами.

Align right edge of selected components (выравнивание по правому краю)

В этом случае для выравнивания используется правый край самого правого компонента из набора выделенных. Компоненты расставляются согласно установленным правилам проектирования, определяющим минимальное расстояние между компонентами.

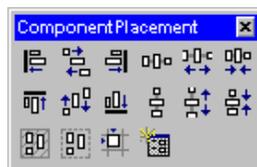


Рис. 6.54. Панель инструментов интерактивного размещения компонентов.

Align vertical centers of selected components (центрирование по вертикали)

В этом случае выделенные компоненты будут выстроены в один вертикальный столбец. Выравнивание будет произведено по центрам компонентов относительно одного, запрос на выбор которого будет выдан после запуска команды.

Align top edge of selected components (выравнивание по верхнему краю)

В этом случае для выравнивания используется верхний край самого верхнего компонента из набора выделенных. Компоненты расставляются согласно установленным правилам проектирования, определяющим минимальное расстояние между компонентами.

Align bottom edge of selected components (выравнивание по нижнему краю)

В этом случае для выравнивания используется нижний край самого нижнего компонента из набора выделенных. Компоненты расставляются согласно установленным правилам проектирования, определяющим минимальное расстояние между компонентами.

Align horizontal centers of selected components (центрирование по горизонтали)

В этом случае выделенные компоненты будут выстроены в одну горизонтальную строку. Выравнивание будет произведено по центрам компонентов относительно одного, запрос на выбор которого будет выдан после запуска команды.

Инструменты для равномерного распределения компонентов

Make horizontal spacing equal for selected components (равномерное распределение по горизонтали)

В данном случае выделенные компоненты будут равномерно распределены по горизонтали. Границами служат самый левый и самый правый компоненты выделения. Вертикальная позиция не изменяется.

Increase horizontal spacing for selected components (увеличение расстояния по горизонтали)

Расстояние между точками привязки компонентов по горизонтали увеличивается равномерно согласно установленному шагу сетки размещения компонентов по оси X.

Decrease horizontal spacing of selected components (уменьшение расстояния по горизонтали)

Расстояние между точками привязки компонентов по горизонтали уменьшается равномерно согласно установленному шагу сетки размещения компонентов по оси X.

Make vertical spacing equal for selected components (равномерное распределение по вертикали)

В данном случае выделенные компоненты будут равномерно распределены по вертикали. Границами служат самый верхний и самый нижний компоненты выделения. Горизонтальная позиция не изменяется.

Increase vertical spacing of selected components (увеличение расстояния по вертикали)

Расстояние между точками привязки компонентов по вертикали увеличивается равномерно согласно установленному шагу сетки размещения компонентов по оси Y.

Decrease vertical spacing of selected components (уменьшение расстояния по вертикали)

Расстояние между точками привязки компонентов по вертикали уменьшается равномерно согласно установленному шагу сетки размещения компонентов по оси Y.

Инструменты перемещения компонентов

Move to Rectangle (переместить компоненты в заданную прямоугольную область)

Данная команда перемещает все выделенные компоненты в пределы прямоугольника, задаваемого с помощью двух углов. Выравнивание компонентов будет произведено как по вертикали, так и по горизонтали.

Move to Room

Данная команда перемещает все выделенные компоненты в пределы заданной области размещения (**Room**). Выравнивание компонентов будет произведено как по вертикали, так и по горизонтали.

Move to Grid

Данная команда перемещает компоненты в ближайшие узлы сетки размещения компонентов. Шаг этой сетки по осям X и Y задается на вкладке **Options** диалогового окна **Document Option**. Заблокированные компоненты не перемещаются.

Автоматическое размещение компонентов

Размещение компонентов является наиболее критическим процессом на стадии разработки печатной платы. Усилия, затрачиваемые на трассировку платы, а также суммарная стоимость ее изготовления сильно зависят от качества размещения элементов.

Процедура размещения подразумевает определение точного положения и ориентации на плате каждого компонента, размера платы и области, в которой допускается размещение компонентов. Наиболее важной целью процесса размещения после установки на плату всех компонентов так, чтобы не нарушались заданные правила проектирования, заключается в том, чтобы упростить работу по трассировке или, в некоторых случаях, просто сделать процедуру трассировки возможной.

Итак, основными целями размещения компонентов на плате являются:

- Установка всех компонентов.
- Не допущение нарушения правил проектирования.
- Расстановка компонентов так, чтобы была возможность выполнить процесс трассировки.

- Удовлетворение требований по сборке и тестированию платы. Для обеспечения этих требований рекомендуется всегда консультироваться с конструктором оснастки платы.

Выбор программы автоматического размещения

Редактор печатных плат пакета Protel 99 SE имеет доступ к двум программам автоматического размещения:

- **Cluster Placer** (программа пакетного размещения) на основе связанности компонентов группирует их в пакеты и размещает эти пакеты по геометрическому принципу. Алгоритмы работы этой программы таковы, что лучше всего ее применять в проектах с малым количеством компонентов (менее 100).
- **Global Placer** (программа глобального размещения) для размещения компонентов использует статистический алгоритм, направленный на минимизацию длин линий связи. Данный алгоритм рекомендуется применять в проектах, где количество компонентов превышает 100.

Задание границ платы для установки компонентов

Перед запуском программы **Global Placer Server** необходимо определить участки платы, где допускается размещение компонентов. Это выполняется посредством нанесения рисунка на слое **Keep Out**. Этот рисунок определяет границу, в пределах которой должны быть размещены все компоненты.

Имеется возможность определить некоторые области на плате, которые не будут заняты компонентами. Для этого с помощью линий, залитых фигур, дуг и полигонов на слое **Keep Out** нужно создать специальные запрещенные зоны (**keep-out**). Более подробно этот вопрос описан в разделе *Описание печатной платы*.

Перед выполнением программы автоматического размещения необходимо с помощью команды меню **Edit » Origin » Reset** восстановить состояние абсолютного начала координат вместо возможно установленного текущего. Это является важным моментом, поскольку программы автоматического размещения используют абсолютное начало координат в качестве опорной точки и могут разместить компоненты вне созданной пользователем сетки, связанной с текущим относительным началом координат.

Настройка и запуск программы пакетного размещения Cluster Placer

Программа пакетного размещения Cluster Placer руководствуется правилами проектирования, установленными на вкладке **Placement** диалогового окна **Design Rules**. Подробно эти правила описаны в разделе *Определение требований к проекту печатной платы*.

После определения этих правил с помощью команды меню **Tools » Auto Place** вызывается диалоговое окно **Auto Place**. Далее для запуска программы пакетного размещения необходимо выбрать опцию **Cluster Placer** и нажать кнопку **OK**.

Настройка программы глобального размещения Global Placer

Программа глобального размещения Global Placer несмотря на сложность ее реализации проста в использовании. Все, что нужно знать, это некоторые наиболее общие факты о проекте. Не требуется специального ее конфигурирования, так как система настроена на обеспечение оптимальной производительности широкого спектра возможных вариантов плат.

Для настройки программы Global Placer необходимо выполнить команду меню **Tools** » **Auto Place**, которая вызывает диалоговое окно **Auto Place**. Далее нужно выбрать опцию **Global Placer** и установить опции программы глобального размещения, как показано ниже.

Group Components (группирование компонентов)

Если эта опция включена, программа Global Placer перебирает все находящиеся на плате компоненты и группирует те из них, которые тесно связаны между собой. Основной критерий при группировании — количество связей между компонентами. Весовой коэффициент, присваиваемый данному критерию, зависит от количества имеющихся у компонентов выводов.

Затем система в пределах каждой группы выполняет относительное размещение. Созданные группы трактуются как "суперкомпоненты", и их внутреннее относительное размещение сохраняется на всем протяжении процесса основного размещения.

Хотя данная опция и является основной, ее использование может привести к обратному результату, если нет достаточно места на плате. Это происходит по причине того, что относительное размещение в пределах групп не изменяется в процессе основного размещения и пространство может теряться при "пристраивании" группы друг к другу.

Rotate Components (вращение компонентов)

Если допускается вращение компонентов, они будут повернуты для получения оптимальной ориентации. Рассматривается четыре варианта ориентации: 90°, 180°, 270° и 360°.

Эта опция должна использоваться с осторожностью, так как вращение посадочных мест может оказывать непосредственное влияние на технологичность разработки. Например, некоторое оборудование для размещения не может оперировать повернутыми компонентами. Включение опции **Rotate Components** также усложняет процедуру размещения и повышает ее продолжительность, особенно в случаях схем высокой плотности. В то же время, с помощью этой опции можно получить очень хороший результат, который обычно требует несколько итераций метода проб и ошибок.

Редактор печатных плат также позволяет после процедуры размещения свободно поворачивать компоненты с точностью 0.001°. Эта операция выполняется с помощью команд меню **Edit** » **Move**, **Edit** » **Paste Array** или посредством использования поля **Rotation** в диалоговом окне **Change Component**.

Automatic PCB Update (автоматическое обновление печатной платы)

Программа Global Placer автоматически передает текущие позиции компонентов на плате в окно редактора всякий раз, когда обновляется статус оптимизации (прибли-

зительно каждые 10 секунд). Также имеется возможность ручного обновления в любое время, пока запущена программа размещения. Выполняется с помощью команды **File » Update PCB**.

Placement Grid (сетка для размещения)

Это сетка, в узлах которой размещаются точки привязки компонентов. Ее шаг обычно устанавливается как доля от общего для всех компонентов шага выводов и/или кратным предполагаемому шагу трассировочной сетки. Редактор печатных плат заставляет инструменты размещения вставлять компоненты в новую сетку, если она по каким-либо причинам будет изменена на более поздних этапах проектирования (команда меню **Tools » Align Components » Move To Grid**).

Power Nets (цепи питания)

Данная опция выполняет две функции:

1. Цепи, которые определены в полях **Power Nets**, больше не принимаются во внимание алгоритмом размещения, что может существенно ускорить процесс размещения. В каждом текстовом поле можно задать несколько цепей питания. Для этого необходимо разделять имена цепей пробелом (максимальное число символов в строке не должно превышать 28).
2. Для связывания шунтирующих конденсаторов с большими компонентами необходимо определить имена цепей, которые эти конденсаторы шунтируют. Например, если в проекте в качестве цепей питания и заземления используются цепи **VCC** и **GND**, в первое поле нужно ввести **VCC**, во второе — **GND**. Программа глобального размещения сделает попытку связать двухвыводной компонент, который соединен параллельно указанным цепям питания (**VCC** и **GND**), с большими компонентами (14 и более выводов).

Запуск программы глобального размещения Global Placer

После конфигурирования программы глобального размещения Global Placer можно запускать саму программу. Для этого достаточно нажать кнопку **OK**.

Окно программы Global Placer

Сервер программы Global Placer отображает в своем окне развитие процесса размещения. Здесь учитываются как компоненты, так и keep-out области.

Главное меню программы Global Placer очень короткое. Оно включает следующие пункты: **File**, **View**, **Window** и **Help**. Так как эта программа имеет собственные структуры данных, то база данных печатных плат не изменяется в процессе размещения. Для передачи данных о текущем размещении компонентов в редактор печатных плат используется команда меню **File » Update PCB**. Такая передача позволяет периодически переходить в окно редактора для слежения за качеством размещения, а также позволяет использовать инструмент **Density Map** для проверки возможности трассировки проекта.

Окно программы размещения имеет собственную строку состояния, в которой отображается следующая информация:

Пройденное время

Время, пройденное с момента запуска процесса размещения.

Оптимизация

Процесс размещения занимает 70 циклов. Первые 40-50 циклов проходят очень быстро, так как большинство выполняемых здесь перемещений принимаются без проблем. Однако, далее для удовлетворения требований цикла количество перемещений возрастает. Это означает, что по мере продвижения к завершающему этапу, циклы становятся все более медленными. Оптимизация имеет отношение к проценту законченности относительно "идеального" образца расхода времени.

Число перемещений

Количество перемещений, отображаемых в строке статуса, — это общее число перемещений компонента на новую позицию, которое выполняет система для улучшения способности платы к трассировке.

◆ В процессе размещения на плате появляются небольшие фиолетовые квадраты. Размер этих квадратов отражает плотность соединения в данной области платы.

Результаты размещения

По мере продвижения процесса размещения к оптимальному решению количество выполняемых перемещений уменьшается. Необязательно доводить этот процесс до полного завершения, его в любой момент можно остановить с помощью команды меню **File » Close**. Перед закрытием окна программы размещения появится запрос на обновление платы.

Советы по улучшению результатов автоматического размещения

Предварительное размещение компонентов

До запуска программы автоматического размещения пользователь имеет возможность самостоятельно расположить некоторые компоненты и затем во избежание их перемещения установить в диалоговом окне **Components** атрибуты **Locked** для этих компонентов.

Кроме компонентов, которые должны быть размещены в определенных местах платы, такие как краевые разъемы, радиаторы или группы аналоговых компонентов, существуют компоненты, не имеющие ограничений на их размещение, но которые желательно размещать вручную. Примером могут служить микросхемы памяти, предварительная ручная установка которых может облегчить размещение других компонентов.

Использование Keep-out областей

Имеется возможность определить некоторые области на плате, которые не будут заняты компонентами. Такие зоны могут потребоваться вокруг разъемов или в местах расположения механических соединений. Эти области создаются на слое **Keep Out** с помощью линий, залитых фигур, дуг и полигонов.

Автоматическое размещение и большие цепи

Наличие больших цепей может значительно повлиять на скорость и качество работы программы автоматического размещения. Причина этого кроется в том, что вычисления, затрачиваемые на перестановку цепей, по экспоненциальному закону зависят от размера цепи. Исходя из практического опыта можно сделать вывод, что большие цепи питания и заземления могут играть незначительную роль в общем процессе размещения. Таким образом, можно улучшить размещение, указав программе те цепи, которые она должна проигнорировать в процессе своей работы.

Для этого программа Cluster Placer требует определения цепей в **Nets to Ignore Design Rule** на вкладке **Placement**, расположенной в диалоговом окне **Design Rules**.

Для программы Global Placer необходимо определить цепи в области **Power Nets** диалогового окна **Auto Place**.

Важно помнить, что программа автоматического размещения компонентов — это эффективный инструмент, но он не может являться полноценной заменой профессиональных навыков и опыта разработчика. Небольшая помощь разработчика, например, предварительное размещение наиболее критичных компонентов и их блокирование, а также задание реалистичных сеток для размещения компонентов и разного рода зазоров, может значительно помочь процессу автоматического размещения.

Инструменты, используемые после процедуры автоматического размещения

Результатом работы программы Global Placer является плата, на которой относительное расположение компонентов подобрано оптимальным образом. Благодаря такому алгоритму работы Global Placer часто создает платы, которые далеки от идеала. Например, после завершения процедуры размещения некоторые компоненты все еще могут перекрываться или выравнивание некоторых компонентов может требовать небольшой доработки. Интерактивные инструменты размещения представляют собой специальные средства для облегчения процесса наведения порядка на плате. Информация по применению интерактивных инструментов размещения приведена ранее в данной главе в подразделе *Интерактивное размещение компонентов*.

Что такое связанность и топология

Средства электронного проектирования представляют собой платформу, при помощи которой проект любого радиотехнического устройства можно превратить из абстракции, витающей в сознании разработчика, непосредственно в файлы для производства печатной платы. Использование мощности и эффективности компьютера позволяет быстро и точно провести проект через различные этапы разработки.

По мере прохождения этих этапов проект хранится в различных форматах. Проект начинается с принципиальной схемы — набора компонентов, связанных друг с другом проводниками. Заканчивается проект печатной платой изготовленной в соответствии с технологическими файлами, сгенерированными средствами проектирования печатных плат.

Суть проекта на всех этих этапах — это информация о компонентах и о том, как они соединяются друг с другом.

Существует механизм для перевода проекта из стадии редактирования принципиальной электрической схемы в стадию разработки печатной платы, который извлекает информацию о компоненте и его связях из принципиальной электрической схемы и переносит ее на рабочую область редактора печатных плат. Переносимая информация о компоненте содержит: позиционное обозначение компонента; его номинал (если есть); тип корпуса (топологическое посадочное место). Информация о связях компонента передается как набор цепей, где каждая цепь — это список выводов компонентов, электрически соединенных друг с другом.

В редакторе печатных плат разработчик превращает информацию о компонентах и их связях в их реальное расположение на печатной плате с физическими связями. Этот процесс включает размещение компонентов в границах платы при наличии других механических ограничений и превращение всех логических соединений цепей в физические связи, в соответствии с электрическими параметрами, необходимыми для этих соединений.

Каким образом передается информация о компонентах и их связях

Система Protel 99 SE имеет мощную утилиту синхронизации проекта, которая значительно облегчает передачу проектной информации из принципиальной электрической схемы на печатную плату и обратно.

Синхронизатор проводит анализ как электрической схемы, так и печатной платы и выявляет все отличия. При запуске синхронизатора разработчику надо выбрать направление синхронизации включением одной из двух опций **Update PCB from Schematic** (обновить плату исходя из изменений в схеме) или **Update Schematic from PCB** (обновить схему исходя из изменений в плате). После этого синхронизатор попытается обновить данные в указанном документе в соответствии с данными из ис-

ходного документа. Вся информация обо всех проделанных изменениях, а также обо всех затруднениях на этой операции записывается в файл отчета синхронизатора.

Подробная информация по использованию синхронизатора приведена в разделе *Передача информации о схеме в редактор печатных плат*.

Как отображается информация о связях

Редактор печатных плат, при загрузке в него синхронизатором информации о компонентах и связанности, показывает соединения между выводами компонентов каждой цепи в виде последовательности тонких линий. Линия, соединяющая два вывода одной цепи, называется маршрутом (**From-To**), то есть линией от одного вывода до другого вывода.

Рисунок или схема расположения маршрутов **From-To** называется топологией цепи. Для цепей, не подвергавшихся предварительной трассировке (без задания топологии пользователем), редактор печатных плат проводит маршруты **From-To** по кратчайшему расстоянию между выводами в соответствии с заданным расположением компонентов.

Для предварительно полностью или частично трассированных цепей маршруты **From-To** проводятся от вывода компонента до ближайшего проводника цепи. При этом они отображаются пунктиром, напоминая разработчику о необходимости провести проводник между этими двумя точками, чтобы обеспечить необходимые связи (рис. 6.55).

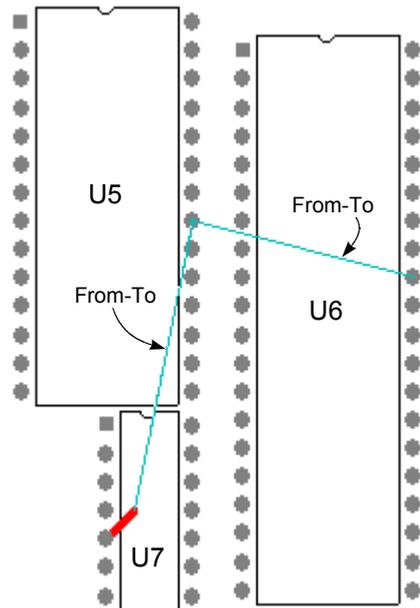


Рис. 6.55. Цепь с одним неразведенным и одним частично разведенным соединением.

Топология цепи задается или правилами проектирования, расположенными на вкладке **Routing**, или определением стационарных маршрутов **From-To**. Ниже в этом разделе будет рассматриваться топология на основе стационарных маршрутов. Подробная информация о правилах проектирования, используемых при трассировке платы, приведена в разделе *Определение требований к проекту печатной платы*.

Топологии цепей

Для всех цепей отображаются линии связи, проходящие между двумя любыми принадлежащими одной цепи выводами. Расположение или структура этих линий связи называется топологией цепей. По умолчанию редактор печатных плат проводит ли-

нии связи для каждой цепи по кратчайшему расстоянию, поэтому такая топология называется кратчайшей. Имеются также другие типы топологии цепей.

Разработчик может переопределить топологию цепи по различным причинам. Например, высокоскоростные печатные платы требуют минимальных отражений сигнала. Для таких плат, с целью получения высоких скоростей, применима топология последовательного соединения выводов (цепочка), в которой вывод источника сигнала находится в начале, а вывод нагрузка — в конце цепи. Другим примером может являться необходимость соединить все земляные выводы цепи в одной общей точке. Такая топология называется "звезда".

Способы изменения топологии цепи

В редакторе печатных плат задать конкретную топологию цепи можно двумя способами. Первый способ — задать топологию для всей цепи сразу (на уровне цепи). Второй — задать топологию индивидуально для каждой пары выводов цепи при помощи маршрутов **From-To** (на уровне маршрутов **From-To**). Если проект печатной платы содержит цепи, требующие определенной топологии, то для таких цепей следует задействовать правила проектирования из вкладки **Routing**. Подробная информация по этому поводу приведена в подразделе *Примеры использования правил проектирования*. Если разработчик желает вручную задать топологию всей цепи или отдельной ее части, необходимо для них создать специальные маршруты **From-To**.

Пользовательские маршруты From-To

Возможность полного управления топологией редактором печатных плат предоставляется посредством функции описания пользовательских маршрутов **From-To**. Такой маршрут предписывает редактору печатных плат провести линию связи от одного заданного вывода к другому заданному выводу.

Разработчик может задать в цепи один, несколько или все без исключения маршруты. При создании пользовательских маршрутов только для части цепи остальные будут расположены по кратчайшему пути.

Помимо задания определенного типа топологии маршруты **From-To** служат объектом для применения правил проектирования. Благодаря им, можно применять специфические правила проектирования не только ко всей цепи, носящей определенное имя, но и к ее части. Например, все дорожки цепи, за исключением одного участка, должны иметь ширину 25 тысячных дюйма, специфический же участок должен иметь ширину 40 тысячных дюймов. Более подробно это описано в подразделе *Примеры использования правил проектирования*.

Создание маршрута From-To

Маршруты **From-To** создаются при помощи собственного редактора **From-To Editor** (рис. 6.56), вызываемого командой меню **Design » From-To Editor**.

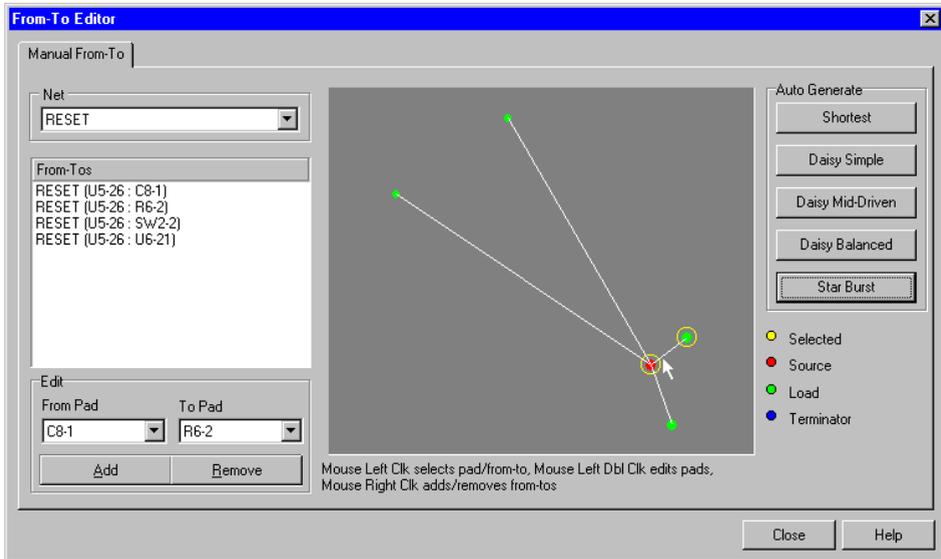


Рис. 6.56. Создание и удаление маршрутов в редакторе From-To Editor.

После того, как появилось окно редактора маршрутов **From-To**, необходимо выбрать цепь, с которой необходимо работать. Все принадлежащие ей выходы отображаются в графическом окне справа. Уже существующие линии связности видны как тонкие линии, проходящие между парами выводов, они отображаются в списке **From-To**, расположенным под названием цепи. Под графическим окном располагаются подсказки по работе с маршрутами.

Автоматическая генерация маршрутов From-To

Для автоматической генерации маршрутов в редакторе существует ряд специальных кнопок на поле **Auto-Generate**. По нажатию одной из них для выбранной цепи создается набор маршрутов согласно типу выбранной топологии:

Shortest (минимальная длина)

По умолчанию редактор печатных плат располагает связи в цепи таким образом, чтобы суммарная длина связей была наименьшей. Нажатие этой кнопки приводит к удалению маршрутов, сформированных автоматически или пользователем, и предписывает редактору печатных плат расположить все маршруты по кратчайшему пути.

Daisy-Simple (простая цепочка)

Эта топология соединяет все узлы в цепочку один за другим. Порядок следования узлов в цепочке рассчитывается так, чтобы минимизировать общую длину. Если заданы начальная и конечная контактные площадки (**source** и **terminator**), то остальные располагаются по принципу минимизации общей длины. Для назначения начальных

или конечных контактных площадок необходимо выполнить их редактирование. Если задано несколько начальных (или конечных) контактных площадок, то они соединяются вместе размещаются на конце цепочки.

Daisy-Mid Driven (цепочка с началом в центре)

При этой топологии начальный узел (или узлы) размещается в центре цепочки, остальные узлы делятся пополам и размещаются по разные стороны от начального узла. При этом необходимо задать два конечных узла, которые помещаются на концах цепочки. При назначении нескольких начальных узлов они соединяются вместе и размещаются в центре цепочки. Если не определено точно два конечных узла, то используется топология простой цепочки **Daisy-Simple**.

Daisy-Balanced (сбалансированные цепочки)

При такой топологии все множество узлов делится на одинаковые по числу узлов цепочки, общее количество которых равно числу конечных узлов. Затем эти цепочки соединяются с начальным узлом по схеме "звезда". Несколько начальных узлов соединяются вместе.

Star ("звезда")

При топологии такого типа каждый узел подсоединяется непосредственно к начальному узлу. Если заданы конечные узлы, они подсоединяются после каждого узла. При наличии нескольких начальных узлов они соединяются вместе, как в топологии типа **Daisy-Balanced**.

Отображение связей между выводами

Для того, чтобы работа конструктора при разработке топологии была более продуктивной, редактор печатных плат обладает возможностью выборочно показывать или прятать соединительные линии между выводами. В меню следует выбрать команду **View » Connections**, которая имеет следующий набор опций:

Цепь (Net)

Для выбранной цепи скрываются или отображаются все маршруты. При выборе этой опции курсор принимает вид перекрестия. Если известно расположение контактной площадки цепи, то для активизации действия опции следует щелкнуть на ней левой кнопкой мыши. В противном случае можно щелкнуть по свободному месту и прочитать в появившемся окне название цепи. Если разработчик не уверен или не знает название цепи, то ему следует нажать на клавишу **?** и нажать кнопку **OK**. На экран выведется список загруженных цепей.

Цепи компонента (Net Components)

Эта опция скрывает или отображает маршруты всех цепей, принадлежащих выбранному компоненту.

Все линии связности (All)

Эта опция скрывает или отображает маршруты всех загруженных, но не разведенных цепей.

◆ Для больших цепей с множеством узлов или большим количеством примитивов (проводников, переходных отверстий и так далее), принадлежащих цепи, автоматический анализ связности может оказаться слишком медленным. Чтобы отключить анализ связности для отдельно взятой цепи, ее необходимо скрыть. После того как пользователь с помощью команды **View » Connections » Hide Net** сделал цепь невидимой, она перестает быть видимой и для программы проверки связности.

Работа со списками соединений

Во время работы над проектом может возникнуть ситуация, когда пользователю потребуется изменить список соединений. Эта операция выполняется в диалоговом окне **Netlist Manager**, вызываемом командой меню **Design » Netlist Manager**. В нижней части окна расположена кнопка **Menu**, с помощью которой можно выполнить следующие действия.

Удалить или добавить контактную площадку к цепи

Чтобы удалить или добавить контактную площадку к цепи, выберите нужную цепь в списке **Nets in Class** и выполните на ней двойной щелчок левой кнопкой мыши. Откроется диалоговое окно **Edit Net**, в котором нужно сделать любые изменения с помощью кнопок, расположенных в центре окна.

Update Free Primitives From Component Pads (обновить топологию по списку соединений)

Эта опция используется в случаях, когда изменения производились в редакторе принципиальных схем. После передачи таких изменений в редактор печатных плат, программа проверки правил проектирования может выявить определенные нарушения, которые возникают в случае несоответствия цепей и выводов компонентов. При выборе этой команды все примитивы устанавливаются в состояние, когда они не подключены ни к одной цепи, после чего их подключение к цепям устанавливается исходя из списка соединений.

Заметим, что данная процедура производится абсолютно со всеми контактными площадками. Поэтому если имеются площадки, соединенные проводником в старой версии, но принадлежащие разным цепям в новой версии, рекомендуется удалить соединение для упрощения работы программы.

Export Netlist from PCB (экспорт списка соединений из редактора печатных плат)

Данная команда выполняет экспорт внутреннего списка соединений в отдельный файл. Под внутренним списком соединений подразумевается тот, который отражается в данный момент в окне **Netlist Manager**.

Create Netlist from Connected Copper (создание списка соединений по существующей топологии)

В этом случае система проанализирует все проводники, существующие на всех без исключения слоях, и создаст по ним список соединений. Если в имени контактной площадки присутствует имя цепи, то оно присваивается всей цепи.

Compare Netlists (сравнение списков соединений)

После выбора этой команда система запросит пользователя указать два сравниваемых списка. Сравнение цепей производится на уровне общих узлов, а все несоответствия заносятся в специальный отчет.

Compare Netlist File to Board (сравнение списка соединения из файла с существующей платой)

При выполнении данной команды производится сравнение внутреннего списка соединений со списком из внешнего файла. Данная опция используется для сравнения внутреннего списка соединений, полученного после проверки правил проектирования, и списка сгенерированного в редакторе принципиальных схем.

Изменение атрибутов цепи

Как и у контактных площадок, проводников и сквозных отверстий, у цепи есть собственный набор атрибутов. Для их редактирования необходимо перевести панель управления редактора печатных плат из режима **Browse** в режим **Nets**, отметить цепь и нажать кнопку **Edit**. Появится диалоговое окно **Change Net**, в котором можно отредактировать атрибуты выбранной цепи. Для настройки доступны следующие атрибуты:

Color (цвет) — Для каждой цепи при загрузке устанавливается цвет, заданный по умолчанию. При необходимости его можно изменить здесь.

Hide (скрыть) — При помощи этого атрибута можно выборочно скрыть маршруты заданной цепи или множества цепей в режиме глобального редактирования.

◆ На вкладке **Layers** диалогового окна **Document Options** есть опция управления отображения слоя **Connect**. Если опция выключена, то линии связи не показываются, а значение атрибута **Hide** игнорируется.

Идентификация цепей

Чтобы получить полную информацию о любом объекте на печатной плате, в том числе и проводнике, необходимо подвести к нему указатель мыши. Вся информация, например, физические размеры или имя цепи, будут отображены в строке состояния. Выключение строки состояния производится на вкладке **Display** диалогового окна **Preferences**.

Ручная трассировка печатной платы

Трассировка печатной платы — это процесс перевода логических соединений в физические. В качестве физических соединений могут выступать печатные проводники, сквозные отверстия, контактные площадки, дуги, области заливки, полигоны и слои питания. Однако, наиболее часто используемыми из них являются печатные проводники и переходные отверстия.

Редактор печатных плат системы Protel 99 SE обладает рядом сервисных функций, специально приспособленных для того, чтобы ускорить процесс перевода логических соединений в физические. В их число входят:

Интеллектуальная ручная трассировка

При размещении проводник, обеспечивающий электрическое соединение, в нужном месте разработчику нет необходимости вести дорожку строго по пути следования линии соединения. Можно довести проводник до другого вывода цепи или создать T-образное соединение. После завершения прокладки проводника цепь будет проанализирована, а виртуальные линии соединения автоматически добавлены или удалены в соответствии с текущим расположением печатных проводников.

Электрическая сетка

Для облегчения точного размещения электрических объектов, таких как проводники или переходные отверстия, редактор печатных плат имеет электрическую сетку. Электрическая сетка задает область, внутри которой любой электрический объект (будь то проводник или переходное отверстие) будет притягиваться к другому электрическому объекту. Электрическая сетка отменяет действие сетки **Snap Grid**, что облегчает соединение объектов, расположенных вне этой сетки.

Размещение объектов без нарушений правил проектирования

В редакторе печатных плат есть такой режим трассировки, при котором разработчик может расположить примитивы только таким образом, чтобы не нарушались правила проектирования, задающие зазоры между различными объектами. Это свойство позволяет поводить максимально плотную трассировку.

Интерактивная проверка правил проектирования

Большинство правил проектирования проверяется во время интерактивной трассировки. Поэтому чтобы контролировать их выполнение надо просто включить нужные правила на вкладке **On-line Tab** диалогового окна **Design Rule Check**, вызываемого командой меню **Tools » Design Rule Check**, после чего нарушения будет показываться сразу по обнаружению.

Автоматическое удаление замкнутых петель проводников

Уже существующие на плате проводники можно быстро перерезать. Новый сегмент проводника просто проводится в нужном месте, а избыточные проводники удаляются автоматически.

Режим прокладки проводников с предсказанием

Этот режим расположения задает режимы излома размещаемых проводников. Проводники могут быть изогнуты плавно по дуге или иметь излом под углом 45°. В любом из этих режимов предсказание можно использовать для прогнозирования размещения следующего сегмента проводника и аккуратного завершения текущего.

Автоматическая вставка переходных отверстий

Во время трассировки переход с одного сигнального слоя на другой осуществляется горячими клавишами, + и -. Переходные отверстия при этом вставляются автоматически.

Как редактор печатных плат работает со связями во время трассировки

Редактор печатных плат системы Protel 99 SE является средой проектирования, управляемой связностью схемы. Это означает, что редактор отслеживает и управляет связями в списке соединений на всех этапах трассировки платы.

Рассмотрим цепь **RD**, показанную на рисунке и соединяющую вывод 5 микросхемы **U7**, вывод 32 микросхемы **U6** и вывод 21 микросхемы **U5**. На принципиальной схеме эта цепь представлена в виде проводников, соединяющих указанные три вывода. После того как список цепей загружен в редактор печатных плат, цепь подвергается анализу и создаются две виртуальные линии связи (маршруты **From-To**), отображаемые в виде тонких линий на слое Connection, как показано на рисунке 6.57.

Задача конструктора состоит в том, чтобы преобразовать эти два маршрута **From-To**, расположенных на слое **Connection** в реальные проводники на сигнальных слоях. Всякий раз, когда разработчик прекращает процесс прокладки проводника, редактор печатных плат проверяет всю цепь, чтобы определить, какие ее участки разведены, а какие нет. В тех местах цепи, где проводники не проведены, вставляются новые линии связи, обеспечивающие связность цепей на плате.

Редактор печатных плат автоматически отслеживает статус завершенности трассировки обрабатываемой цепи, поэтому, разработчик может прокладывать проводники, не учитывая расположение этих линий связи. Например, конструктор начал вести проводник от вывода 5 **U7** и решил вести его к выводу 32 **U6**, вместо 21 **U5**. После за-

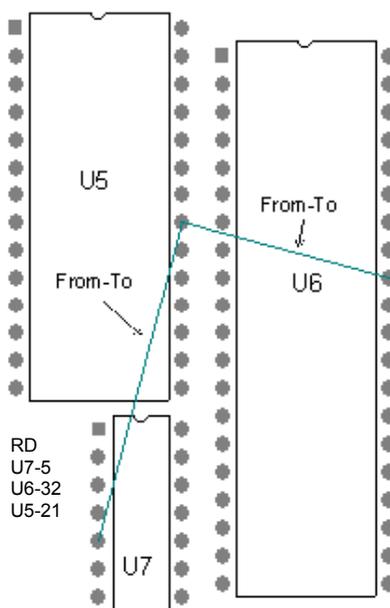


Рис. 6.57. Так отображается неразведенная цепь.

вершения прокладки этого сегмента проводника редактор печатных плат анализирует целостность цепи **RD** и добавляет новую линию связи **From-To** от несоединенного вывода к ближайшей точке только что образованной подцепи, как показано на рисунке ниже.

Напомним, что основная задача разработчика состоит в том, чтобы перевести соединения из виртуальных линий связи в физические проводники, а задача редактора печатных плат — отслеживать ход трассировки и по мере необходимости обновлять эти линии.

Такая методика имеет два ярко выраженных преимущества. Во-первых, можно провести проводник к любому примитиву, при этом вовсе необязательно вести проводник для соединения двух выводов строго по виртуальным линиям связи. Редактор печатных плат по мере прокладывания проводника автоматически добавляет и удаляет линии связи. Во-вторых, цепь всегда остается неразрывной, то есть ее не возможно разделить на две не соединенные друг с другом части. При удалении сегмента проводника редактор печатных плат обнаруживает разрыв и незамедлительно добавляет новую виртуальную линию связи для восстановления связности цепи (рис. 6.58).

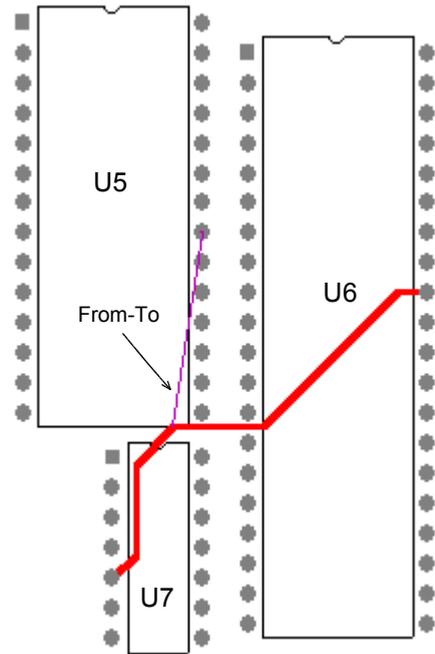


Рис. 6.58. После анализа не до конца разведенной цепи редактор печатных плат добавляет маркер разорванной цепи (Broken Net Marker).

При анализе цепи и добавлении линий связи редактор печатных плат руководствуется заданной топологией цепи. Как указывалось раньше, для всех цепей по умолчанию принимается топология, при которой длина всех маршрутов минимальна (**Shortest**). Для цепей с такой топологией линии связи проводятся между двумя ближайшими сегментами проводника, принадлежащего одной цепи.

Если для данной цепи назначена другая топология, то линии связи добавляются согласно ее требованиям. В этом случае линии связи прорисовываются пунктирной линией, говорящей о том, что для сохранения данной топологии, следует провести проводник между указанными двумя точками.

Топология цепи или маршрута определяется правилом проектирования **Routing Topology Rule**. Более подробная информация по этому вопросу приведена в подразделах *Работа со списками соединений* и *Правила проектирования, учитываемые при трассировке (Routing)*.

◆ Если проект состоит из очень больших цепей с множеством узлов, или же цепь содержит огромное количество примитивов (проводники, переходные отверстия и так далее), то анализ целостности цепи потребует большего времени. Чтобы выключить контроль связности конкретной цепи, ее необходимо скрыть. Это можно сделать, выполнив команду меню **View » Connections » Hide Net**. Скрытая цепь становится недоступной и для модуля проверки связности.

Подготовка к трассировке

Подготовка проекта к трассировке является одной из самых важных этапов процесса конструирования. Ниже приведены рекомендации по оптимизации процесса трассировки.

Настройка сеток

Традиционно печатные платы всегда разводились в некоторой заданной сетке. Шаг сетки рассчитывался из условий быстрого и точного размещения объектов на печатной плате без возможности нарушить правила проектирования. Например, печатная плата, на которую устанавливались компоненты с выводами шагом 100 мил (2.54 мм) в сквозные отверстия, должна была трассироваться в сетке 25 мил (0.635 мм). Такая сетка позволяла бы прокладывать параллельные проводники шириной 12 мил с зазором 13 мил, а между двумя выводами микросхемы всего один такой проводник.

Большое разнообразие в типах топологических посадочных мест, а также использование как дюймового, так и метрического шага, привело к затруднениям в точной настройке стандартной сетки так, чтобы она удовлетворяла всем требованиям проекта и компонентов. Это стало главным недостатком систем проектирования печатных плат, основанных на сетке размещения.

Редактор печатных плат системы Protel 99 SE имеет ряд сервисных функций, помогающих конструктору преодолеть эти ограничения. Сюда входят: возможность использовать электрическую сетку, позволяющую одному электрическому объекту присоединиться к горячей точке (**hot-spot**) другого электрического объекта, даже если он находится вне текущей сетки размещения; размещение проводников с предсказанием, показывающее разработчику, как наилучшим образом завершить трассировку текущего сегмента проводника и проложить следующий; автоматическая проверка правил проектирования. Благодаря всем этим функциям, бессеточный интерактивный трассировщик позволяет провести проводник быстро и точно к любому объекту и в любую точку чертежа печатной платы.

Несмотря на пугающие сложностью названия, работать с этими функциями очень просто. Если проект разработчика не подходит ни под один стандартный набор настроек, или же требования по плотности размещения проводников и переходных отверстий не выполняются с помощью модели на базе сетки размещения, следует установить еще меньший шаг сетки snap grid, например 5 или 1 мил. Как трассировать плату в этом режиме, рассказывается в этой главе ниже.

Более подробная информация по установке параметров сетки **Snap Grid**, сетки размещения **Component Grid** и электрической сетки **Electrical Grid** приведена в подразделе *Система сеток* раздела *Настройка редактора чертежей печатных плат*.

Размещение компонентов в сетку

Для увеличения числа доступных каналов трассировки количество контактных площадок компонентов, попадающих в узлы сетки **Snap Grid**, должно быть как можно большим. Проверить, находятся ли компоненты в сетке, можно с помощью команды меню **Edit** » **Select** » **Off Grid Pads** (горячие клавиши **S, G**). Привязку всех компонентов к узлам сетки размещения можно выполнить с помощью команды меню **Tools** » **Interactive Placement** » **Move To Grid** (горячие клавиши **I, G**), после чего на экране появится диалоговое окно, позволяющее пользователю установить параметры сетки.

Проверка плотности трассировки

Редактор печатных плат имеет функцию просмотра карты плотности трассировки (**Density Map**), которая помогает перед разводкой предварительно определить размеры шага сетки, ширину проводников, переходных отверстий, число слоев и так далее. Ее вызов производится командой меню **Tools** » **Density Map**, после чего печатная плата будет раскрашена в различные цвета в зависимости от плотности трассировки. Зеленым цветом на ней отображаются "холодные" области с низкой плотностью трассировки, а красным — "горячие" области с высокой плотностью. При наличии больших зон красного цвета с высокой плотностью можно, проанализировав текущее размещение компонентов, попытаться удалить их, вручную отодвинув компоненты друг от друга или изменив шаг сетки и ширину проводников. Если это не помогает, необходимо увеличить число слоев трассировки.

Включение слоев трассировки

Редактор печатных плат позволяет работать с 32 сигнальными слоями (верхний, нижний и 30 внутренних) и 16 слоями питания и заземления. Если конструкция проекта требует наличия внутренних сигнальных слоев, а конструктор намеревается применить глухие и скрытые переходные отверстия, то сигнальные слои должны быть разбиты на пары. Описание стека слоев приведено в разделе *Описание печатной платы* и подразделе *Переходные отверстия* раздела *Объекты редактора печатных плат*.

Настройка правил проектирования

Редактор печатных плат является средой, управляемой согласно установленным правилам проектирования. То есть, во время размещения какого-либо объекта на чертеже печатной платы система проверяет, какие из ранее заданных правил проектирования применимы к этому объекту и не нарушаются ли они.

При создании новой печатной платы для нее создается набор правил проектирования по умолчанию. Ширина проводников при этом равна 10 мил, зазор между проводниками — 10 мил, диаметр (стиль) переходного отверстия — 50 мил и приоритетное направление трассировки: горизонтальное на верхнем слое, вертикальное на нижнем.

Перед началом процесса трассировки, необходимо правильно установить эти правила. Подробное описание правил проектирования приведено в разделе *Определение требований к проекту печатной платы*.

Трассировка печатной платы вручную

Разводка проводников редактором печатных плат, непрерывно отслеживающим связность цепей, очень проста. При размещении прямого или дугообразного проводника, переходного отверстия или области металлизации редактор печатных плат отслеживает связность цепей и соответствующим образом обновляет виртуальные линии связи.

Режим размещения проводника на плату включается командой меню **Place » Interactive Routing**. Если щелкнуть левой кнопкой мыши по объекту, которому соответствует имя некоторой цепи, размещаемый проводник принимает ее имя, становясь составной частью этой цепи. Если щелкнуть указателем мыши по линии связи (маршруту **From-To**), то он переместится к ближайшей контактной площадке. После прокладки сегмента проводника линия связи переместится к его концу.

После выхода из режима размещения проводников редактор печатных плат проверяет целостность цепи и обновляет виртуальные линии связи. Линии связи могут подсоединяться к проложенному проводнику в любой точке, а не только на его конце и точках излома. Редактор, основываясь на заданной топологии данной цепи, добавляет линии связи между всеми участками цепи, которые не соединены проводниками. По умолчанию выбирается топология с минимальной длиной проводников, поэтому линии связи будут располагаться по кратчайшему пути между отдельными точками не подсоединенных сегментов цепи. Более подробная информация по этому вопросу приведена в разделах *Что такое связность и топология* и *Правила проектирования, учитываемые при трассировке (Routing)*.

Ширина проводника задается правилом проектирования **With Constraint**. При создании нового проекта печатной платы, для него формируется набор правил с установленными по умолчанию значениями, где ширина всех проводников на плате задается равной 10 мил. Для отдельных цепей можно задавать свою ширину проводников. Подробная информация об этом приведена в разделе *Определение требований к проекту печатной платы*.

Изменение параметров проводников и переходных отверстий во время трассировки

Изменение параметров проводников и переходных отверстий может быть выполнено непосредственно во время размещения с помощью клавиши **TAB**. Появится диалоговое окно **Interactive Routing**, где можно изменить ширину проводника, диаметр переходного отверстия и его кольца металлизации (рис. 6.59). Сделанные здесь изменения отразятся на рекомендуемых значениях (параметр **Preferred**) правил проектирования **Width Constraint** и **Routing Via Style**. Если устанавливаемое значение выходит за установленные рамки, то оно автоматически ограничивается верхним или нижним значением.

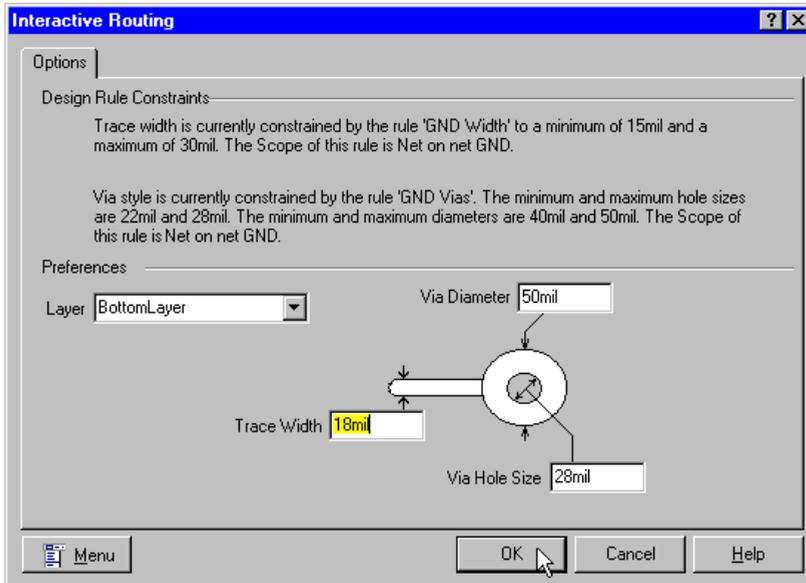


Рис. 6.59. Настройка интерактивного трассировщика.

Горячие клавиши, используемые во время ручной трассировки

Для ускорения процесса трассировки предназначены следующие горячие клавиши:

- Клавиша **BACKSPACE** удаляет последний излом проводника;
- Клавиша **~** переключает слои трассировки в процессе размещения проводников;
- Клавиша **/** используется для добавления переходного отверстия с текущего слоя на внутренний слой питания или заземления.
- Клавиша **SPACEBAR** переключает режим размещения **Start** или **End**. Одновременное нажатие **SHIFT + SPACEBAR** изменяет режим размещения проводников;
- Одновременное нажатие **SHIFT+SPACEBAR** циклически выбирает линии связи, соединенные с данной контактной площадкой.
- Клавишей **END** обновляется содержимое экрана.
- Удержание клавиши **CTRL** временно отключает электрическую сетку.
- Комбинация клавиш **SHIFT+E** последовательно включает или выключает электрическую сетку.
- Удержание клавиши **ALT** временно переключает режим **Avoid Obstacle** в режим **Ignore Obstacle**.

- Одновременное нажатие клавиш **SHIFT+R** циклически переключает три режима интерактивной трассировки.

Размещение проводников с предсказанием

Редактор печатных плат включает в себя модернизированную функцию предварительной прорисовки, работающей во время размещения предыдущего сегмента проводника. Контур сегмента проводника, конец которого подсоединен к указателю мыши, до окончательного размещения прорисовывается пунктиром. Сегмент проводника, находящийся между предварительно прорисованным сегментом и последним размещенным называется текущим размещаемым сегментом. Пример работы функции показан на рисунке 6.60.

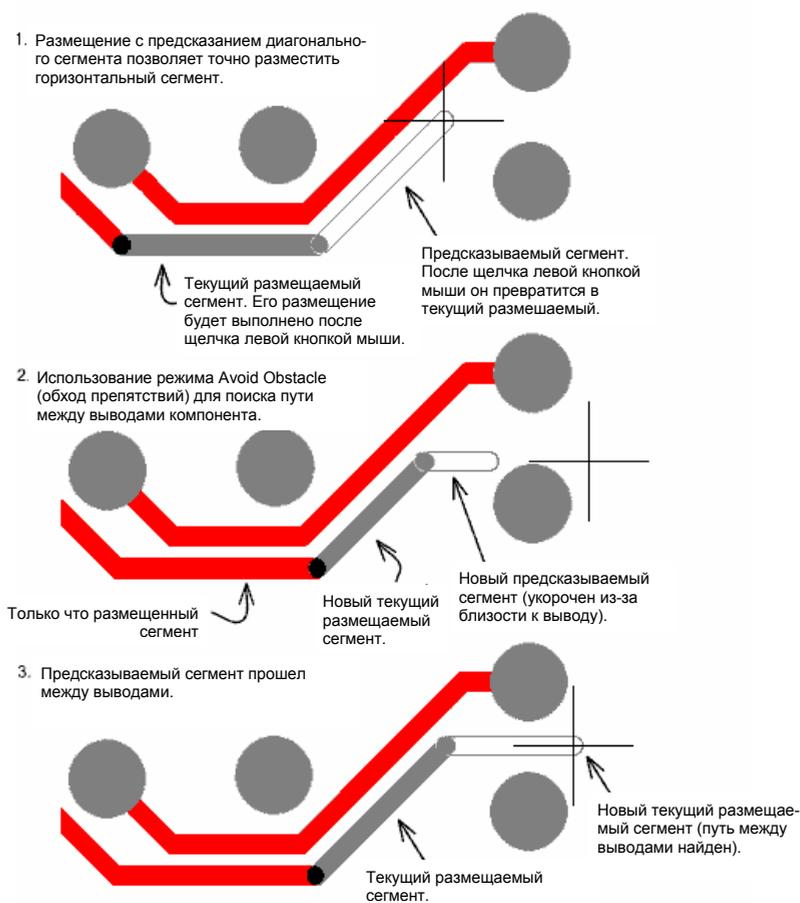


Рис. 6.60. Возможность предварительной прорисовки предсказываемого сегмента позволяет правильно разместить текущий сегмент.

Механизм предварительной прорисовки очень полезен для быстрого и точного расположения текущего проводника рядом с уже существующими объектами на плате, а также для планирования местоположения следующего сегмента проводника. Когда разработчик щелчком мыши фиксирует текущий сегмент, его конечная точка будет расположена точно в том месте, откуда должен начинаться следующий сегмент.

При пользовании механизма предсказания можно заметить, что конец следующего сегмента проводника не всегда остается прикрепленным к указателю мыши. Это говорит о том, что он уклоняется от объектов, принадлежащим другой цепи. Это свойство позволяет располагать примитивы с учетом правил проектирования, определяющих зазоры между ними, как это показано на предыдущем рисунке (шаг 2). Сначала указатель мыши был перемещен в точку справа от контактных площадок, а предварительно прорисованный сегмент усечен до места, где начинает срабатывать установленное правило проектирования. Как только указатель мыши передвинули в точку, в которой создались условия для прокладывания предсказываемого сегмента между контактными площадками без каких-либо нарушений, сегмент удлинился до этой точки. Различные режимы трассировки будут более подробно рассмотрены в следующем разделе данной главы.

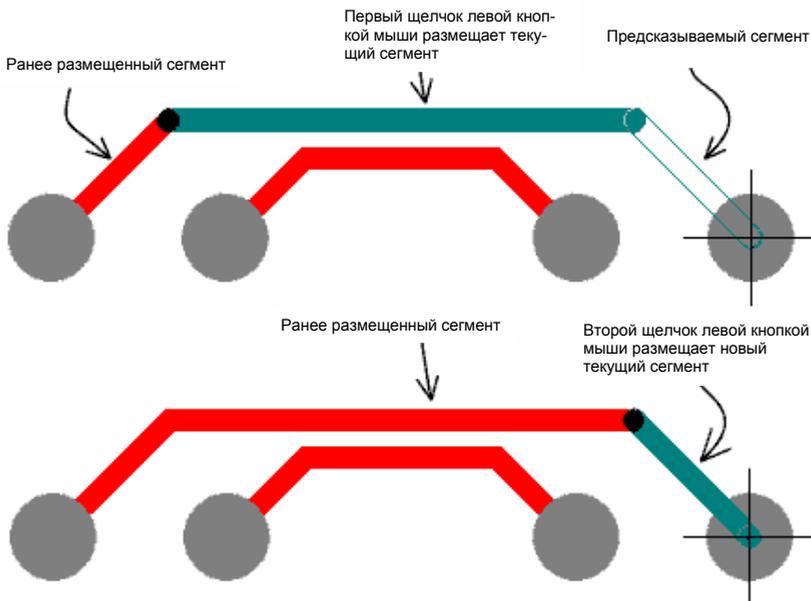


Рис. 6.61. Прокладка диагональных проводников с предсказанием.

Рассмотрим другой пример ручной трассировки, приведенный на рисунке 6.61. Предположим, что необходимо провести проводник параллельно другому проводнику и затем по диагонали вниз до контактной площадки. Раньше идеальное место окончания горизонтального сегмента и начала диагонального определялось методом проб и ошибок. Механизм предсказания позволяет просто навести указатель мыши

на намеченную контактную площадку, затем первым щелчком левой кнопкой мыши завершить горизонтальный сегмент, а вторым — следующий диагональный сегмент.

Следует помнить, что показанный пунктиром сегмент — это лишь предсказываемый, а не прокладываемый в данный момент сегмент. Если при щелчке левой кнопкой мыши сегмент не фиксируется, то это означает, что вы вероятнее всего пытаетесь проложить предсказываемый сегмент. Это случается, когда режим размещения находится в положении **End** вместо режима **Start** и наоборот. Переключение между режимами размещения **Start** и **End** производится клавишей **SPACEBAR**.

Режимы интерактивной трассировки

Редактор печатных плат имеет три различных режима интерактивной трассировки: игнорирование препятствий; обход препятствий; расталкивание препятствий. Эти режимы определяют поведение проводника по отношению к уже существующим объектам на плате. Все три режима работают совместно с механизмом удаления петель проводников.

Игнорирование препятствий (Ignore Obstacle)

В этом режиме проводник можно прокладывать в любом месте окна редактора печатных плат. При включенной автоматической проверке правил проектирования (**Tools** » **Design Rule Check** и **Tools** » **Preferences**), если размещаемый проводник нарушает одно из них, то он и мешающий объект подсвечиваются, сигнализируя об ошибке. Данный режим особенно полезен при повторной разводке платы, когда необходимо создать временное нарушение правил, которое при последующей трассировке будет устранено.

Обход препятствий (Avoid Obstacle)

В этом режиме при разводке проводников создать нарушения невозможно. Если при прокладке проводника указатель мыши доходит до места, где создаются условия для нарушения установленных правил проектирования, сегмент проводника ограничивается позицией, в которой все правила проектирования еще соблюдаются. Режим уклонения от препятствий делает задачу трассировки предельно простой, то есть разработчику нет необходимости тщательно отслеживать расположение проводников. Редактор печатных плат разрешает размещать их только в тех местах, где отсутствуют нарушения правил проектирования. Отметим, что это самый предпочтительный режим для ручной трассировки.

- | |
|--|
| ◆ Удержание клавиши ALT временно переключает режим Avoid Obstacle в режим Ignore Obstacle . |
|--|

Прокладывание проводников внутри полигона (Plowing Through Polygons)

Если в диалоговом окне **Preferences** включена опция **Plow Through Polygon**, пользователь может прокладывать проводники поверх существующих полигонов в интерактивном режиме **Avoid Obstacle**.

После завершения прокладки проводника полигон будет автоматически перезалит, если опция **Polygon Repour** в диалоговом окне **Preferences**. Если здесь установлено ограничение на число примитивов в поле **Threshold**, в случае превышения этого числа будет выдан запрос на подтверждение выполняемых действий.

Расталкивание препятствий (Push Obstacle)

В этом режиме прокладываемый проводник отталкивает со своего пути встретившиеся и мешающие проводники. Убрать с пути нельзя только непереключаемые объекты, например, переходные отверстия, контактные площадки и заблокированные проводники. При возникновении ситуации, когда выталкиваемый проводник не может двигаться дальше, так как ему мешает непереключаемый объект, система переключается в режим игнорирования препятствий. Режим расталкивания идеально подходит для повторной трассировки платы. При работе в этом режиме из-за большого количества изменений на экране может возникнуть необходимость обновить его содержимое, что выполняется нажатием клавиши **END**.

◆ Во время интерактивной трассировки режимы размещения проводников циклически переключаются клавишами **SHIFT + R**.

Повторная трассировка платы

Повторная трассировка платы является обычным этапом процесса конструирования, которая наверняка понадобится после автоматической трассировки, а также если на плату были добавлены новые компоненты или были изменены корпуса уже существующих. Редактор печатных плат имеет мощные функции, помогающие в процессе повторной трассировки платы, например, расталкивание препятствий и удаление петель проводников. Обе этих функции конфигурируются на вкладке **Options** диалогового окна **Preferences**, вызываемого командой меню **Tools » Preferences**.

Ниже приведена последовательность действий при повторной разводке существующего проводника:

1. Выполните команду меню **Place » Interactive Routing**.
2. Чтобы начать процесс трассировки, щелкните левой кнопкой мыши по проводнику или контактной площадке.
При этом цепь, к которой этот объект принадлежит, подсветится, а указанный сегмент будет захвачен указателем мыши.
3. Проложите проводник по новому пути, расталкивая, при необходимости, существующие дорожки, и соедините его со старым проводником в любом месте или на контактной площадке.
4. Завершить процесс трассировки нажатием правой кнопки мыши или клавиши **ESC**.

Если, объединившись со старым, новый проводник образует петлю, то редактор автоматически ликвидирует ее (рис. 6.62). При повторной разводке также могут быть созданы новые переходные отверстия, причем старые будут автоматически удалены.

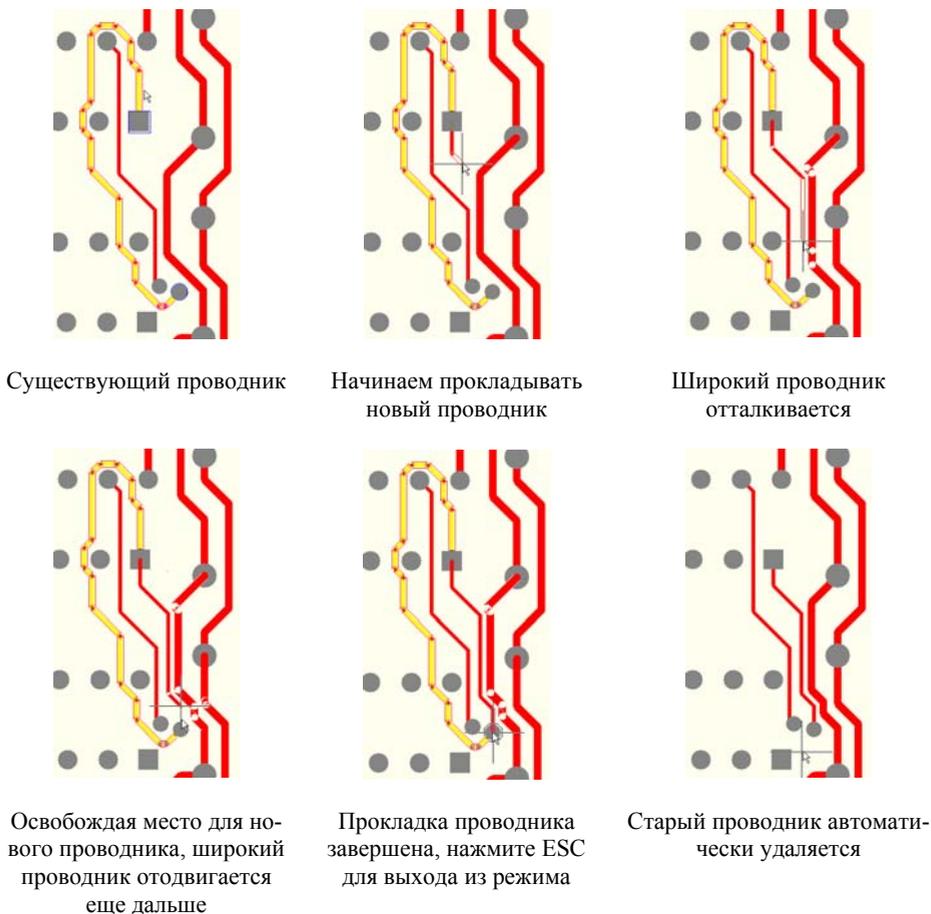


Рис. 6.62. Повторная интерактивная трассировка цепи.

◆ Удаление петель проводников и режим расталкивания препятствий включаются на вкладке **Options** диалогового окна **Preferences**.

Внутренние слои питания и заземления

Внутренние слои питания представляют собой особые внутренние слои со сплошной металлизацией. Редактор печатных плат системы Protel 99 SE поддерживает до 16 слоев питания. Если проект построен на основе списка цепей, каждому из них можно назначить определенное имя цепи. После разбиения слоя на несколько изолированных друг от друга областей, каждой из них может быть назначена своя цепь.

Выводы компонентов могут соединяться со слоем питания либо непосредственно, либо с использованием теплового барьера. Тепловой барьер используется во время пайки для сокращения передачи тепла от вывода компонента к области металлизации, с которой он соединен электрически. Редактор печатных плат позволяет задать форму теплового барьера индивидуально для каждой отдельной или для всех одновременно контактных площадок, соединяемых со слоем питания.

Для соединения с внутренними слоями питания контактных площадок выводов компонентов, предназначенных для поверхностного монтажа, применяются специальные средства. Планарные контактные площадки автоматически помечаются как подсоединенные к этому слою. Для выполнения физического соединения автотрассировщик использует так называемый "стрингер" — короткий проводник с переходным отверстием на соответствующий внутренний слой питания с соединением непосредственно или с помощью теплового барьера.

Назначение цепи для внутреннего слоя питания

Чтобы назначить внутреннему слою питания какую-либо цепь необходимо:

1. Выполнить команду меню **Design » Layer Stack Manager**, после чего появится диалоговое окно **Layer Stack Manager**.
2. Добавить новый внутренний слой питания или заземления к существующему стеку слоев (если он не был создан ранее), выполнить щелчок левой кнопкой мыши на слое, под которым будет размещаться новый слой, и нажать кнопку **Add Plane**. Следует учитывать, что, как правило, печатные платы имеют четное число слоев металлизации, поэтому при добавлении нового слоя пользователю может потребоваться добавить еще один сигнальный или внутренний слой.
3. Выполнить двойной щелчок левой кнопкой мыши на слое и назначить ему нужную цепь или присвоить новое имя.
4. Закрыть диалоговое окно **Layer Stack Manager**.

Все линии связи для этой цепи исчезнут. При наведении указателя мыши на контактную площадку, принадлежащую данной цепи, он будет принимать вид небольшого перекрестия. Причем для непосредственного соединения он будет иметь вид ×, а для соединения с тепловым барьером он будет похож на †.

◆ Параметры соединения контактной площадки или переходного отверстия с внутренним слоем питания или заземления описываются правилом проектирования **Power Plane Connect Style**, описание которого приведено в разделе *Определение требований к проекту печатной платы*.

Выводы, не присоединяемые к слою питания

Выводы, не присоединяемые к внутреннему слою питания, изолируются от него с помощью свободной от меди области. Размеры этой области задаются правилом проектирования **Power Plane Clearance**, описанным в разделе *Определение требований к проекту печатной платы*.

Соединение переходных отверстий с внутренним слоем питания

Аналогично контактным площадкам, переходные отверстия автоматически соединяются с внутренним слоем питания, если ему назначена та же цепь. Соединение осуществляется согласно установленному для этой цепи правилу проектирования **Power Plane Connect Style**.

Если пользователь не хочет соединять переходные отверстия к внутреннему слою питания, то в соответствующих правилах проектирования **Power Plane Connect Style** необходимо установить стиль **No Connect**. Более подробная информация об этом приведена в подразделе *Примеры использования правил проектирования* раздела *Определение требований к проекту печатной платы*.

Просмотр внутреннего слоя питания

Содержимое внутренних слоев питания отображается в негативном изображении. Металлизация под всеми размещенными на них объектами будет удалена, в то время как незаполненные области на чертеже будут иметь сплошную металлизацию. Чтобы посмотреть, каким образом выводы компонентов соединяются с внутренним слоем питания, необходимо сделать следующее:

1. Включить только нужный слой питания, и слои **pad holes layer** и **multi layer**.
2. Щелкнуть на вкладке слоя питания в нижней части окна редактора печатных плат.
3. При необходимости, обновить содержимое экрана, нажав клавишу **END**.

Пример соединения контактной площадки с внутренним слоем питания с использованием теплового барьера приведен на рисунке 6.63. Контактные площадки с непосредственным соединением показываются одной черной точкой в месте расположения отверстия контактной площадки.

Создание разделенных слоев питания

Когда разработчику требуется разделить один внутренний слой питания между несколькими цепями, необходимо разбить его на несколько изолированных областей. Как правило, цепь, имеющая наибольшее количество соединенных с ней выводов, первой ассоциируется со слоем питания и забирает себе большую его часть. Остальная часть слоя отводится другим цепям, назначенным на этот слой. Изолированная

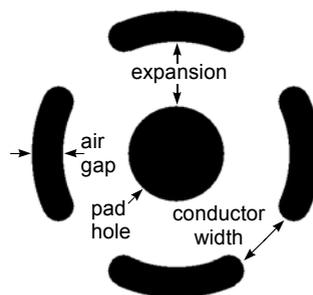


Рис. 6.63. Параметры теплового барьера.

область, принадлежащая другой цепи, задается при помощи специальных линий (границ раздела), охватывающих все выводы этой цепи. Если какой-либо вывод не попадает в эту область, к нему проводится виртуальная линия связи, говорящая о том, что этот вывод должен быть подключен к данной цепи через проводник на сигнальном слое.

Так как внутренние слои питания отображаются в негативе, металлизация из-под линий раздела областей будет удалена, чем обеспечивается изоляция между разными цепями.

Перед тем, как создавать разделенные слои питания рекомендуется прочесть подраздел *Рекомендации по созданию разделенных слоев питания*. Чтобы разделить внутренний слой питания следует:

1. Соединить цепь, имеющую максимальное количество выводов компонентов, с внутреннем слоем питания, как описано в разделе *Назначение цепи для внутреннего слоя питания*.
2. Щелчком левой кнопки мыши на вкладке слоя, расположенной в нижней части окна редактора печатных плат, сделать данный внутренний слой текущим.
3. Командой меню **Design » Internal Planes** вызвать диалоговое окно **Split Planes**.
4. Для разделения слоя на области нажать кнопку **Add**. На экране появится диалоговое окно **Split Plane**, где надо установить ширину изолирующей границы, слой и подключаемую цепь.
5. После настройки всех параметров окна **Split Plane**, нажать кнопку **OK**. Диалоговое окно исчезнет, а указатель мыши примет вид перекрестия.
6. Щелкая мышью обозначить границы области, не забыв для создания замкнутого контура вернуться в начальную точку.

После того как контур замкнется, вновь появится диалоговое окно **Internal Planes**, в котором в списке **Split Plane** появится новая область. Просмотреть новую область можно, щелкнув на ней левой кнопкой мыши.

◆ Переключение режимов размещения границ изолированных областей производится нажатием клавиши **SPACEBAR**. Допускается прорисовка границ с помощью дуг.

Разделение слоя питания для подключения двух и более цепей

Если слой питания предназначен для подключения двух и более цепей, то нужно продолжать разбивать слой для создания других дополнительных областей. При необходимости граничные линии различных областей могут накладываться друг на друга. Желательно делать это с небольшим перекрытием, чем обеспечивается больший зазор между областями.

При создании одной изолированной области меди внутри другой, внешняя область должна охватывать внутреннюю так, как это показано на рисунке 6.64.

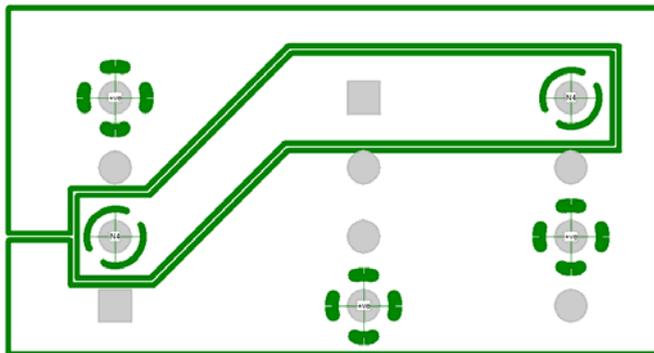


Рис. 6.64. Пример разделенного слоя питания одна цепь находится внутри другой цепи.

Рекомендации по созданию разделенных слоев питания

Чтобы легче находить контактные площадки, которые следует охватить границами одной из областей внутреннего слоя питания необходимо:

1. Включать минимально возможное количество слоев: **Keep Out**, **multi layer**, другие необходимые механические слои и внутренний слой питания, с которым предстоит работать в данный момент.
2. Установить на вкладке **Show/Hide** диалогового окна **Preferences** режим отображения контактных площадок **Draft mode**.
3. Включить на вкладке **Options** диалогового окна **Preferences** опции **Highlight In Full** и **Use Net Color Highlight**.
4. Задать для каждой цепи свой цвет. Для этого надо на панели управления установить режим просмотра цепей (**Nets**), выбрать цепь из списка и нажать кнопку **Edit** (редактировать), чтобы появилось диалоговое окно **Change Net**.
5. Выполнить команду меню **Edit » Select » Net** затем щелкнуть левой кнопкой мыши по одной из контактных площадок нужной цепи. Повторить то же самое для другой назначенной слою цепи. Для того чтобы одновременно можно было выбрать несколько цепей, на вкладке **Options** диалогового окна **Preferences** должна быть включена опция **Extended Selection**. Теперь эти два набора контактных площадок будут раскрашены в разные цвета, что делает их легко различимыми.
6. Теперь все готово для разделения слоя питания на области. Сначала слою назначается самая большая цепь, затем вырезаются области для других цепей.

Редактирование границ областей на внутренних слоях питания

После того как пользователь определил границы разделенных областей, он может их редактировать, причем изменению могут быть подвержены:

1. Ширина граничной линии (зазор между областями), внутренний слой, на котором будут располагаться разделенные области, а также подключаемые к слою цепи. Чтобы изменить один из этих атрибутов, необходимо двойным щелчком левой кнопкой мыши на границе между областями вызвать диалоговое окно разделения областей **Split Plane**.
2. Расположение граничных линий. Для этого в меню следует выбрать строку **Edit » Move » Split Plane Vertices**. Пользователю будет предложено выбрать полигон (**Choose a Polygon**), так как разделенная область представляет собой всего лишь незаполненный полигон. Появятся метки-манипуляторы, перемещая которые можно отредактировать границы области. Более подробная информация о модифицировании полигонов, а также о том, какие действия необходимо предпринять для редактирования их границ, приведена в пункте *Полигоны* раздела *Объекты редактора печатных плат*.

Создание областей металлизации на сигнальном слое

Одними из наиболее часто встречающихся элементов печатных плат являются области металлизации. Это могут быть заштрихованные заземленные полигоны на печатных платах аналоговых устройств, сплошные полигоны на слоях питания, обеспечивающие протекание больших токов, или сплошные заземленные слои, выполняющие роль электромагнитных экранов.

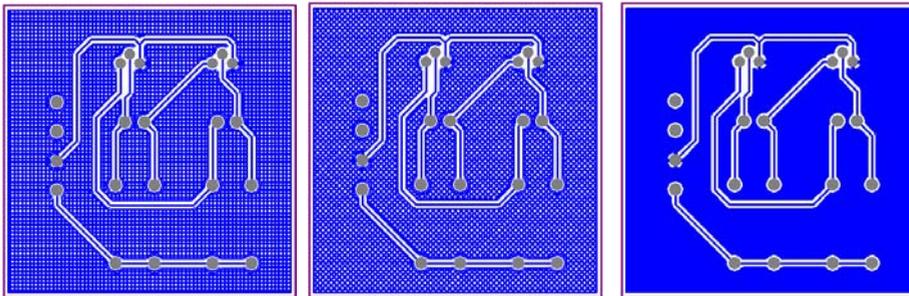


Рис. 6.65. Различные стили заливки полигонов.

Область металлизации может быть создана на любом сигнальном слое посредством задания границ полигона, который впоследствии автоматически будет залит медью в соответствии с заранее определенными правилами проектирования.

Области металлизации могут быть сплошными, а также иметь сетчатый или штрихованный узор. Штриховка может быть вертикальной или горизонтальной, а сетка может быть прорисована под углом 45° или 90° (рис. 6.65).

Для задания границ полигона необходимо выполнить команду меню **Place » Polygon Plane**. Полигон может быть соединен с некоторой цепью. В этом случае все контактные площадки, соединенные с этой цепью и попавшие в рамки полигона, будут автоматически соединены с областью металлизации в соответствии с установками правила проектирования **Polygon Connect Style**. Для того чтобы изменить стиль соединения контактных площадок с металлизацией, необходимо выполнить команду меню **Design » Rules** и в появившемся окне выбрать вкладку **Manufacturing**.

По умолчанию полигон будет иметь сетчатый узор, так как шаг сетки будет превышать ширину штриховых линий. При сплошной заливке шаг сетки делается меньше ширины штриховых линий, в результате чего они сливаются. Более подробная информация о создании и редактировании полигонов приведена в пункте *Полигоны* раздела *Объекты редактора печатных плат*.

Автотрассировка печатных плат

Редактор печатных плат системы Protel 99 SE имеет встроенный высококачественный бессеточный автотрассировщик. Этот модуль подключается к оболочке Design Explorer как отдельный сервер и полностью интегрируется в редактор печатных плат.

Работать с автотрассировщиком чрезвычайно просто. Надо лишь правильно задать определенные правила проектирования и выполнить команду меню **Autoroute » All**. Процесс разводки пользователь может наблюдать в активном окне редактора печатных плат.

Настройка программы автотрассировки

Так как сервер, выполняющий автотрассировку, полностью интегрирован в редактор печатных плат, то и работает он в окне этого редактора. Автоматическая разводка проводников на плате производится согласно установленным правилам проектирования.

Настройка правил проектирования для автотрассировщика

Настройка правил проектирования, используемых программой автотрассировки, настраиваются в диалоговом окне **Design Rules**, вызываемом с помощью команды меню **Design » Rules**, до ее запуска. Автотрассировщик использует подавляющее большинство правил, но не для всех из них можно установить определенную область действия. Подтверждение того, что данное правило используется программой автотрассировки, приводится в нижней части окна **Design Rules**.

Блокирование ранее разведенных проводников

Часто возникает необходимость в предварительной ручной трассировке определенных цепей с последующим запуском программы автоматической трассировки, которая обработает оставшиеся цепи. Блокирование ранее разведенных цепей производится включением опции **Lock All Pre-routes** в диалоговом окне **Autorouter Setup**.

Перед началом автотрассировки на плате не должно быть нарушений установленных правил проектирования. Это означает, что все предварительно разведенные проводники должны удовлетворять заданным для них правилам проектирования. Например, если имеется предварительно разведенные цепи заземления, использующие проводники шириной 20 и 50 тысячных дюйма, тогда правило проектирования **Width Constraint** для цепей заземления должно иметь значение минимальной ширины равное 20 mils и значение максимальной ширины равное 50 mils.

Процедуры автотрассировки

Для настройки процедур автотрассировки необходимо выполнить команду меню **Tools » Autoroute » Setup Autorouter**, после чего появится диалоговое окно **Autorouter Setup** (рис. 6.66).

При отсутствии особых причин не рекомендуется менять установленный по умолчанию набор процедур автотрассировки, который запускает абсолютно все алгоритмы. В описании каждой отдельной процедуры приводятся случаи, когда стандартный набор процедур может быть изменен.

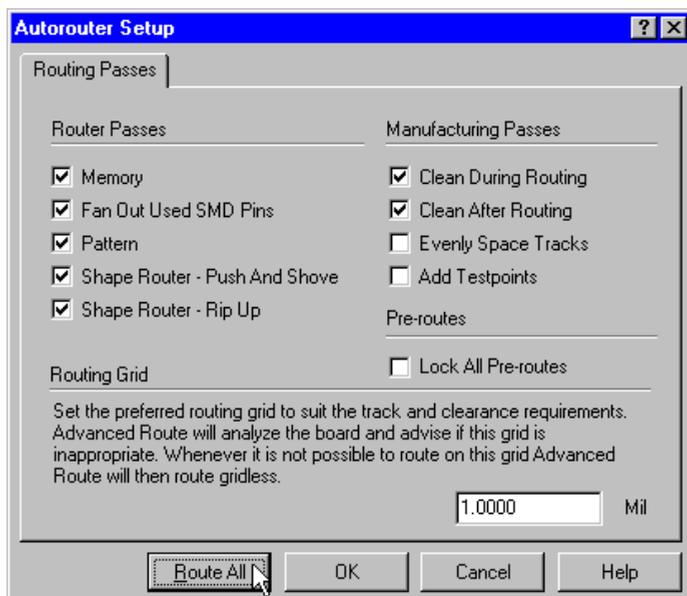


Рис. 6.66. Настройка процедур автотрассировки.

Memory (память)

Эта процедура находит все цепи, связанные с устройствами памяти, или похожие на таковые и использует эвристический (**Heuristic**) алгоритм и метод поиска (**Search**). Рекомендуется всегда включать эту процедуру, даже в случаях, когда на плате нет запоминающих устройств.

Если на разрабатываемой вами плате присутствует настоящий банк памяти, и его местоположение, ориентация и привязка строго оговорены, для оценки его разводки рекомендуется включить только эту процедуру, а все остальные выключить.

Fan Out (веер)

Эта процедура используется для веерообразного расположения переходных отверстий относительно контактных площадок компонента. Это выполняется с помощью отвода от контактных площадок коротких сегментов проводников, на концах которых помещаются переходные отверстия. Данная процедура применяет как эвристический (**Heuristic**) алгоритм, так и метод поиска (**Search**) и должна использоваться при наличии односторонних компонентов на верхнем, нижнем или обоих слоях. Места

неудачного результата работы данной процедуры помечаются небольшой окружностью желтого цвета с символом X в центре.

Насыщенные платы с компонентами для поверхностного монтажа на обоих слоях могут сильно затруднить работу процедуры. В этом случае рекомендуется сначала сделать пробный запуск с включением единственной описываемой процедуры. Если обнаружится, что процедура **Fan Out** не смогла удачно обработать около 10% от общего числа выводов, тогда с большой степенью вероятности можно заключить, что 100% завершение не будет достигнуто. При таком исходе пользователю следует рассмотреть вопрос о перераспределении компонентов в области платы, где работа процедуры закончилась неудачно.

Pattern (шаблон)

Практически к любой плате можно подобрать шаблоны соединений. Успех трассировки этих шаблонов зависит от последовательности выбора соединений во время такой трассировки. Программа трассировки шаблонов имеет набор различных алгоритмов, каждый из которых удобно использовать с тем или иным шаблоном. Программа трассировки шаблонов представляет собой трассировщик методом поиска (**Search**) и ее следует использовать в большинстве случаев.

Push and Shove (вставка и расталкивание)

Данная процедура является основной в программе автоматической трассировки. Она значительно расширена по сравнению с подобными программами других производителей. Процедура умеет вставлять проводники, раздвигая соседние по диагонали, не имеет ограничений на расстояние при отодвигании проводников, может перескакивать через переходные отверстия и контактные площадки, то есть предоставляет новый уровень совершенных и мощных алгоритмов вставки и расталкивания.

Rip Up (разрыв)

Программа автоматической трассировки выполняет доводку платы для полного исключения конфликтов и несоответствий. После завершения процедуры **Push and Shove** (вставки и расталкивания) все еще могут оставаться некоторые несоответствия, которые обозначаются небольшими окружностями желтого цвета. В большинстве случаев успешное выполнение различных трассировочных алгоритмов приводит к устранению таких конфликтов, но бывают ситуации, когда применение всех доступных алгоритмов не дает нужных результатов. Процедура **Rip Up** используется для принудительного разрыва уже проложенных проводников, связанных с конфликтными точками на плате, и повторной их трассировки для устранения конфликтов.

Процедуры улучшения платы для промышленного производства

Эти процедуры равномерно размещают проводники, угловые изгибы проводников и добавляют назначенные тестовые точки в цепи. Если они включены в начальную настройку программы трассировки, тогда их запуск производится по завершении первичных процедур трассировки. В противном случае будет выполнен запуск только первичных процедур, но затем после проверки результатов их работы, можно активировать и запустить данные процедуры.

Clean During Routing u Clean After Routing (процедуры очистки)

Две этих процедуры служат для чистки чертежа печатной платы. Под чисткой подразумевается, удаление лишних изломов или спрямление проводников. Одна из процедур предназначена для работы совместно с имеющимися процедурами автотрассировки, другая запускается после завершения выполнения всех их.

Evenly Space Tracks (равномерное размещение проводников)

Эта процедура используется для равномерного размещения проводников между контактными площадками в следующем случае: параметры трассировки позволяют провести два проводника между контактными площадками микросхемы, но проведен только один, он может быть размещен на 20 тысячных дюйма ближе к какой-либо из контактных площадок. Запуск данной процедуры сдвинет этот проводник в центр промежутка между контактными площадками микросхемы.

Add Testpoints (вставка контрольных точек)

Включение данной процедуры в процесс трассировки предписывает автотрассировщику вставлять контрольные точки для каждой цепи проекта. Перед добавлением новой контактной площадки для контрольной точки программа проверяет все существующие контактные площадки и переходные отверстия данной цепи на предмет возможности их использования в этом качестве. Контрольные точки будут добавляться согласно правилам проектирования Testpoint, прилагаемым к соответствующим цепям. Информация по этому вопросу приведена в разделе *Добавление контрольных точек и каплевидных контактных площадок* данной главы.

Опции программы автоматической трассировки

Программа автотрассировки помимо возможности обработки всей платы имеет несколько других опций.

Autoroute All (автоматическая трассировка всей платы)

Для трассировки всей платы согласно текущим настройкам необходимо выполнить команду меню **Autoroute » All**.

Autoroute Connection (автоматическая трассировка соединений)

Эта опция позволяет выбрать порядок автоматической трассировки от соединения к соединению. Так как при выполнении команды **Autoroute » Connection** используются не все алгоритмы программы автоматической трассировки, то в этом режиме невозможно провести трассировку абсолютно всей платы.

Autoroute Net (автоматическая трассировка цепи)

После выполнения команды меню **Autoroute » Net** необходимо поместить указатель мыши над любым соединением, принадлежащим некоторой цепи, и щелкнуть на нем левой кнопкой мыши. Все соединения данной цепи будут подвергнуты трассировке с использованием всех включенных трассировочных процедур.

Autoroute Component (автоматическая трассировка связей компонента)

После выполнения команды меню **Autoroute » Component** и щелчка левой кнопкой мыши по одному из выводов компонента все соединения, начинающиеся или заканчивающиеся на этом компоненте, будут подвергнуты трассировке. Заметим, что в пределах цепей будут разведены только те маршруты, которые связаны с выбранным компонентом.

Autoroute Area (автоматическая трассировка области)

После выполнения команды меню **Autoroute » Area** и выделения с помощью мыши некоторой области платы все соединения, начинающиеся и заканчивающиеся в ней, будут подвергнуты трассировке.

Программа автоматической трассировки пакета Protel 99 SE — взгляд изнутри

Почему программа автоматической трассировки пакета Protel 99 SE является одной из лучших?

Программа автоматической трассировки пакета Protel 99 SE принадлежит к последнему поколению программ автотрассировки плат в длинной истории развития подобных программ, которая охватывает почти три десятилетия, но имеет некоторые отличия. До выхода в свет пакета Protel 99 SE программы автотрассировки имели две основные цели: выполнить полную разводку платы и выполнить ее в минимальный срок. Автотрассировщик системы Protel 99 SE помимо этих других целей имеет еще одну — обеспечить высокое качество трассировки, что делает ее одной из самых мощных систем проектирования печатных плат для персональных компьютеров.

Исторически сложилось, что большинство профессиональных разработчиков плат придерживаются мнения, что плата, подвергнутая автоматической трассировке, не может достигнуть того классического качества, которое обеспечивает трассировка, выполненная человеком в интерактивном режиме. При этом отмечают большое количество недостатков, которые включают: большое число переходных отверстий; беспорядочное прохождение проводников по плате; вертикальные слои содержат большое количество горизонтальных сегментов и наоборот; проводники распределены неравномерно и после автотрассировки становится затруднительным редактирование платы. Часто возникают замечания о преобладании ортогонального стиля трассировки над диагональным, который опытные разработчики используют подсознательно. Это происходит не только из-за того, что после диагональной трассировки плата имеет более эстетически приятный внешний вид, но по тому, что при таком стиле в некоторой заданной области можно провести большее количество сегментов проводников, чем при ортогональной трассировке. Таким образом, при использовании диагонального стиля повышается вероятность полной трассировки более сложных плат.

Подводя итог, можно сказать, что программа автоматической трассировки пакета Protel 99 SE имеет три основных цели: 100% завершенность, максимально возможная скорость и качество трассировки, приближающееся к качеству профессиональных

разработчиков печатных плат. Для достижения этих целей программа использует как новые технологии, так и старые проверенные временем алгоритмы.

Что такое угловое направление на слое?

Трассировка в специфичном неортогональном направлении на слое многослойной платы — это концепция, применяемая опытными профессиональными разработчиками. Однако, до появления пакета Protel 99 SE она не была реализована ни в одной программе автоматической трассировки.

Простыми словами, трассировка с "направлением на слое" означает упорядочение некоторого соединения на слое, согласно назначенному пользователем направлению, с использованием наклонной линии излома заданного стиля вместо ортогональных изломов.

При таком подходе длина проводника, проходящего по аппроксимирующей гипотенузе, короче, чем длина суммы ортогональных проводников, а также меньше необходимых переходных отверстий. Оба этих фактора являются наиболее ценными при проектировании высокоскоростных логических схем.

Дополнительным преимуществом "углового направления на слое" для любых схем, как высокоскоростных, так и низкоскоростных, является увеличение вероятности полного завершения процесса трассировки.

Выбор используемого направления на слое полностью лежит на разработчике и при необходимости может быть ограничен лишь горизонтальным или вертикальным направлением. Во время упорядочивания при автотрассировке выбираются те соединения, которые наиболее подходят под направление слоя и выполняется их трассировка в выбранном для данного слоя направлении.

Основное перемещение трансформирует длинные сегменты, проходящие по диагонали под углом 45° , с увеличивающимся ортогональным шагом, что приводит к выравниванию первоначального направления к выбранному направлению.

Трассировка в условиях конфликтов

Термин "конфликт" — новое слово в словаре технологии автоматической трассировки, его смысл раскрывается следующим образом. Все трассировочные алгоритмы, входящие в систему Protel 99 SE являются конфликтными, то есть алгоритму не запрещается размещать проводник или переходное отверстие там, где уже расположены другие проводники, переходные отверстия или контактные площадки. Такая ситуация называется конфликтом, правильное разрешение которого и является главной целью программы.

Другими словами, под конфликтом подразумевается нарушение предварительно установленных правил проектирования, которое может заключаться в пересечении сегментов проводников разных цепей, касании или наложении проводников, контактных площадок и переходных отверстий, относящихся к разным цепям.

Когда в некоторой точке платы возникает конфликт, создается маленькая желтая окружность (маркер конфликта) с центром в точке конфликта, которая будет удалена только после окончательного его разрешения. Данная окружность будет служить

сигналом для выполняемых процедур трассировки, что в этом месте платы требуется переработка топологии. Лишь в редких случаях, когда на плате недостаточно места для подавления конфликтной ситуации с помощью функции расталкивания, она сохраняется до конца процесса трассировки.

Общее число конфликтных ситуаций, встреченных по ходу процесса трассировки, отображается в строке состояния программы. Иногда число маркеров конфликтов может не совпадать со значением в строке состояния. Это связано с тем, что маркеры постоянно добавляются и удаляются в процессе выполнения трассировки, а данные строки состояния обновляются только по завершению очередной процедуры.

Добавление контрольных точек и каплевидных контактных площадок

Контрольные точки представляют собой специальные объекты на печатных платах, предназначенные для проверки правильности их изготовления. Критерием контроля служит наличие электрического контакта в проводниках одной цепи и изоляции между проводниками различных цепей до монтажа компонентов.

Контрольная точка служит контактной площадкой для подключения измерительной колодки автоматического контрольного оборудования или щупа прибора при ручном тестировании.

Автоматическое оборудование прикладывает к каждой из контрольных точек заданное напряжение и ток, после чего делает замер токов и напряжений во всех остальных контрольных точках. На ненагруженной (без компонентов) плате проверяется электрический контакт между различными узлами цепей, наличие короткого замыкания и сопротивление изоляции между узлами различных цепей. На смонтированной плате контрольные точки используются для подключения измерительных приборов, контролирующих характеристики конечной схемы.

Процесс вставки контрольных точек делится на следующие этапы:

- Определение конфигурации контрольной точки;
- Определение цепей, требующих добавления контрольных точек;
- Поиск контактных площадок и переходных отверстий, которые могут быть использованы в качестве контрольных точек;
- Добавление контрольных точек в процессе автотрассировки;
- Создание отчета с указанием местоположения все контрольных точек;
- Создание отчета с указанием всех цепей, к которым не удалось добавить контрольные точки.

Что такое контрольная точка?

В качестве контрольной точки может быть использована любая контактная площадка или переходное отверстие. Для этого необходимо просто включить атрибут **Testpoint**

для выбранного объекта с указанием слоя. Можно назначить только верхний (**Top**), только нижний (**Bottom**) или оба этих слоя.

При этом разработчик должен явно осознавать, каким образом будет тестироваться данная плата. Возможно, щупы будут подключаться только с нижней стороны платы, возможно, наоборот, только с верхней. Некоторые цепи может потребоваться проверить с разных сторон.

Задание параметров контактных площадок и переходных отверстий, которые могут быть использованы в качестве контрольных точек, производится с помощью специальных правил проектирования.

Настройка правил проектирования для контрольных точек

Правило проектирования Testpoint Style

Правило **Testpoint Style** определяет требования на физические размеры переходных отверстий и контактных площадок, используемых в качестве контрольных точек (рис. 6.67).

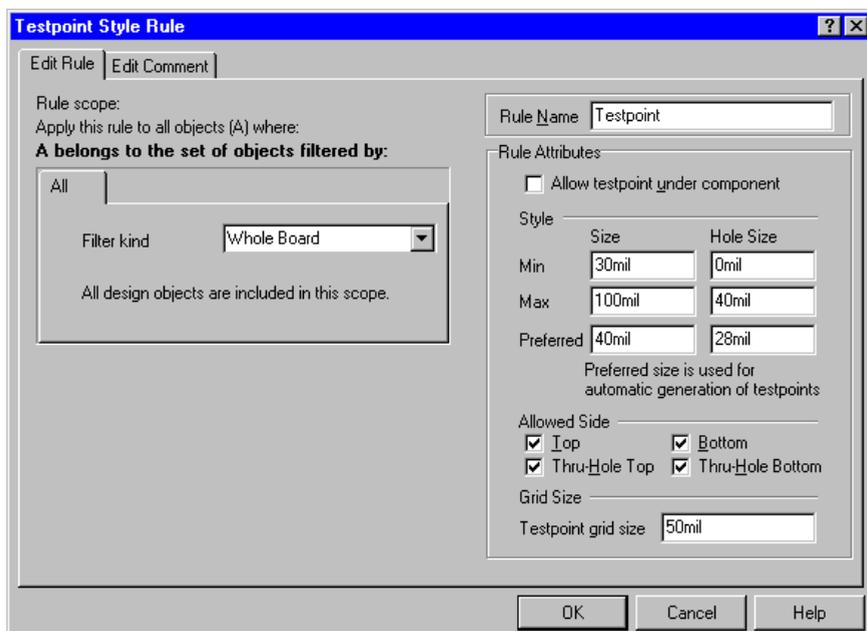


Рис. 6.67. Правило проектирования Testpoint Style определяет стиль переходных отверстий и контактных площадок, используемых в качестве контрольных точек.

Здесь задаются:

Allow testpoint under component — Если данная опция включена, допускается размещение контрольных точек под монтируемым на том же слое компонентом. Это применяется только в случаях, когда плата будет тестироваться только перед монтажом компонентов. На размещение контрольных точек на противоположной стороне платы ограничений не накладывается.

Style — Здесь задаются минимальные, максимальные и рекомендуемые физические размеры отверстия в плате и кольца металлизации контрольных точек. Эти параметры используются функцией поиска контрольных точек **Testpoint Find**, а также при интерактивной или пакетной проверке правил проектирования. Рекомендуемые значения используются программой автотрассировки. Указанный набор параметров полностью описывает размещаемые контрольные точки.

Allowed Side — Опции расположенные в этом поле определяют стиль исполнения контрольных точек (с отверстием или без отверстия), а также слой размещения. Эти опции также используются функцией поиска контрольных точек **Testpoint Find**, а также при интерактивной или пакетной проверке правил проектирования и автотрассировке. Функция **Testpoint Find** и программа автотрассировки использует данные опции в следующем порядке:

- **Bottom** – контактные площадки компонентов для поверхностного монтажа на нижнем слое;
- **Top** – контактные площадки компонентов для поверхностного монтажа на верхнем слое;
- **Bottom thru-hole** – переходные отверстия на нижнем слое со сквозным отверстием;
- **Top thru-hole** – переходные отверстия на верхнем слое со сквозным отверстием;
- **Bottom thru-hole** – контактные площадки на нижнем слое со сквозным отверстием;
- **Top thru-hole** – контактные площадки на верхнем слое со сквозным отверстием;

Все установленные опции обязательно проверяются и учитываются. Опции **Bottom thru-hole** и **Top thru-hole** распространяются и на переходные отверстия, и на контактные площадки. Все они будут перечислены в списке согласно их приоритету.

Grid Size — Здесь задается шаг сетки размещения контрольных точек.

Правило проектирования Testpoint Usage

Данное правило определяет цепи, которые должны иметь контрольные точки. Наборы цепей определяются установкой соответствующей области действия правила проектирования.

Подбор существующих объектов для использования в качестве контрольных точек

Функция **Testpoint Find** позволяет проанализировать существующие контактные площадки и переходные отверстия на возможность использования их в качестве кон-

трольных точек. Запускается эта функция с помощью команды меню **Tools » Find and Set Testpoints**. Используя правила проектирования **Testpoint Usage** и **Testpoint Style**, эта функция проверит параметры всех существующих на плате контактных площадок и переходных отверстий на соответствие установленному стилю **Testpoint Style** для каждой цепи, к которой должны быть добавлены контрольные точки, и соответствующим образом изменит параметры подобранных объектов. По окончании этого процесса появится диалоговое окно с отчетом о количестве контактных площадок и переходных отверстий, используемых в качестве контрольных точек, и сколько из них осталось вставить. Все параметры контрольных точек можно сбросить с помощью команды меню **Tools » Clear all Testpoints**.

Если установленный в системе по умолчанию порядок поиска контрольных точек по каким-либо причинам не соответствует требованиям проекта, рекомендуется запускать эту процедуру несколько раз с разными установками стороны платы.

Добавление контрольных точек во время автотрассировки

Для того чтобы программа автотрассировки автоматически добавляла на плату контрольные точки, необходимо включить в диалоговом окне **Autorouter Setup** опцию **Add Testpoints**. Вставка осуществляется с учетом правил проектирования **Testpoint Style** и **Testpoint Usage** после завершения выполнения всех процедур автотрассировки.

Если плата уже частично разведена и включена опция **Add Testpoints**, то программа автотрассировки сначала проанализирует все существующие на плате контактные площадки и переходные отверстия на предмет возможности использования их в качестве контрольных точек.

Отчет о местоположении контрольных точек

Отчет о местоположении контрольных точек генерируется модулем **CAM Manager** и содержит исчерпывающую информацию о каждой контрольной точке на данной плате. Для создания нового отчета необходимо выполнить команду меню **File » CAM Manager**. Более подробно процесс создания такого отчета описан в разделе *Генерация выходных файлов для производства*.

Отчет о цепях, к которым контрольные точки добавлены не были

Список всех цепей, к которым контрольные точки не были добавлены по каким либо причинам, можно получить с помощью программы проверки правил проектирования DRC, так как невозможность добавления напрямую связана с невыполнением правил проектирования **Testpoint Style** и **Testpoint Usage**. Для генерации такого отчета необходимо включить опцию **Testpoint Usage** на вкладке **Report Tab** диалогового окна **Design Rule Check**, вызываемого командой меню **Tools » Design Rule Check**, опцию **Create Report File** в нижней части этого окна и нажать кнопку **Run DRC** для запуска проверки правил проектирования. По окончании проверки будет сгенерирован отчет, в котором помимо прочей информации будет содержаться список цепей, к которым контрольные точки добавлены не были.

Аналогичным образом, включив опцию **Testpoint Style rule** в диалоговом окне **Design Rule Check**, можно получить отчет о контактных площадках и переходных отверстиях, использованных в качестве контрольных точек.

Удаление всех контрольных точек с платы

Удаление контрольных точек с платы осуществляется выключением атрибута **Testpoint** для всех контактных площадок и переходных отверстий, которое можно выполнить с помощью команды меню **Tools » Clear all Testpoints**.

Создание контактных площадок и переходных отверстий каплевидной формы

Для придания контактным площадкам и переходным отверстиям каплевидной формы, используется команда меню **Tools » Teardrops**, с помощью которой к этим объектам на плате добавляются специальные графические примитивы. В появившемся диалоговом окне **Teardrop Options** следует установить все необходимые параметры, а именно:

All Pads (все контактные площадки) – Добавляет специальные примитивы ко всем контактным площадкам согласно состоянию опций **Selected Objects Only** и **Force Teardrops**.

All Vias (все переходные отверстия) – Добавляет специальные примитивы ко всем переходным отверстиям согласно состоянию опций **Selected Objects Only** и **Force Teardrops**.

Selected Objects Only (только выделенные объекты) – Добавляет специальные примитивы только к выделенным контактным площадкам и переходным отверстиям. Для изменения статуса выделения больших наборов объектов необходимо воспользоваться функциями глобального редактирования.

Force Teardrops – Добавляет специальные примитивы ко всем без исключения контактным площадкам и переходным отверстиям, даже тем для которых это вызывает нарушение установленных правил проектирования.

Create Report (создание отчета)– Включает создание списка все контактных площадок и переходных отверстий, к которым были и не были добавлены специальные графические примитивы.

Верификация проекта печатной платы

Верификация проекта печатной платы является главной и неотъемлемой частью процесса разработки, обеспечивающий и гарантирующий его правильность и завершенность. Процесс верификации должен гарантировать логическое соответствие печатной платы принципиальной электрической схеме и обеспечивать ее физическую работоспособность. То есть электрические объекты, например, проводники, контактные площадки и переходные отверстия должны размещаться на некотором расстоянии друг от друга, предотвращая тем самым замыкание разных цепей.

Редактор печатных плат имеет мощную систему проверки правил проектирования, которая контролирует соответствие проекта требованиям, заданным набором правил проектирования. Проект тестируется на наличие нарушений правил проектирования, таких как недостаточные зазоры, неразведенные цепи, неправильно выбранные ширина и длина проводников, а также других условий, возникающих в процессе трассировки влияющих на изготовление печатных плат.

Редактор печатных плат имеет функцию автоматического контроля правил проектирования (DRC), которая во время трассировки проверяет плату на соответствие включенным правилам проектирования, чем достигается высокая производительность работы конструктора. Для устранения выявленного нарушения необходимо просто передвинуть объект, вызвавший нарушение, а затем продолжить трассировку.

◆ Для подтверждения факта того, что проект выполнен корректно и доведен до конца, пользователю необходим некоторый набор критериев. Редактор печатных плат использует в качестве таких критериев набор правил проектирования, задаваемых в диалоговом окне **Design Rules** и список цепей, загруженный в начальный момент разработки печатной платы. Подробная информация о правилах проектирования приведена в разделе *Определение требований к проекту печатной платы*.

Что представляет собой модуль Design Rule Checker?

Программа Design Rule Checker (DRC) представляет собой высокопроизводительный автоматизированный программный модуль, проверяющий как логическую, так и физическую целостность проекта печатной платы. Использование этого модуля при трассировке для контроля соблюдения минимальных зазоров и отсутствия других нарушений обязательно. Так как редактор печатных плат позволяет в любое время вносить изменения в проект, рекомендуется всегда проводить проверку правил проектирования перед окончательным выводом чертежей.

Интерактивная проверка правил проектирования

Функция проверки правил проектирования в режиме реального времени активизируется на вкладке **Options** диалогового окна **Preferences**. Включение этой функции в режиме ручной трассировки позволяет незамедлительно обнаружить и выделить ошибки в работе. Правила, проверка которых будет производиться в режиме реального времени, настраиваются на вкладке **On-Line** диалогового окна **Design Rule Check**, вызываемого командой меню **Tools » Design Rule Check**.

Если какое-либо правило не доступно (то есть отображается серым цветом), это означает, что пользователь не задал правило этого типа, на основании чего могла бы производиться проверка. Настроить правила проектирования можно с помощью команды **Design » Rule**.

Наличие в проекте больших цепей с полигонами и внутренними слоями питания может резко снизить скорость работы за счет больших затрат времени на проверку связности цепи. Исключить проверку связности для заданной цепи можно, скрыв цепь с помощью команды меню **View » Connections » Hide Net**.

Настройка пакетного режима проверки правил проектирования

Выполните команду меню **Tools » Design Rule Check**, после чего появится диалоговое окно **Design Rule Check** (рис. 6.68).

Пакетный режим проверки правил проектирования имеет следующие опции:

Различные правила проектирования

Здесь на трех полях включаются правила, согласно которым пользователь планирует провести проверку. Описание правил проектирования приведено в разделе *Определение требований к проекту печатной платы*.

Create Violations (высвечивание нарушений)

При включении этой опции места нарушения правил (примитивы) будут выделяться специально заданным цветом. Режим просмотра нарушений можно установить и на панели управления, что особенно полезно для их быстрого нахождения на чертеже и исправления.

Sub-Net Details

Эта опция работает совместно с правилом проектирования **Unrouted Net Rule**. Ее включают в случаях, когда требуется просмотреть все детали неразведенных цепей. Правило проектирования **Unrouted Net Rule** должно включаться только после трассировки всех соединений, так как виртуальная линия связи соединений воспринимается программой проверки как разомкнутая цепь.

Stop When Found XX Violations

Редактор печатных плат автоматически остановит процесс проверки правил проектирования при нахождении заданного числа нарушений.

Create Report File

При включении этой опции автоматически создается файл отчета программы проверки правил проектирования с расширением **.DRC**, который затем открывается текстовым редактором.

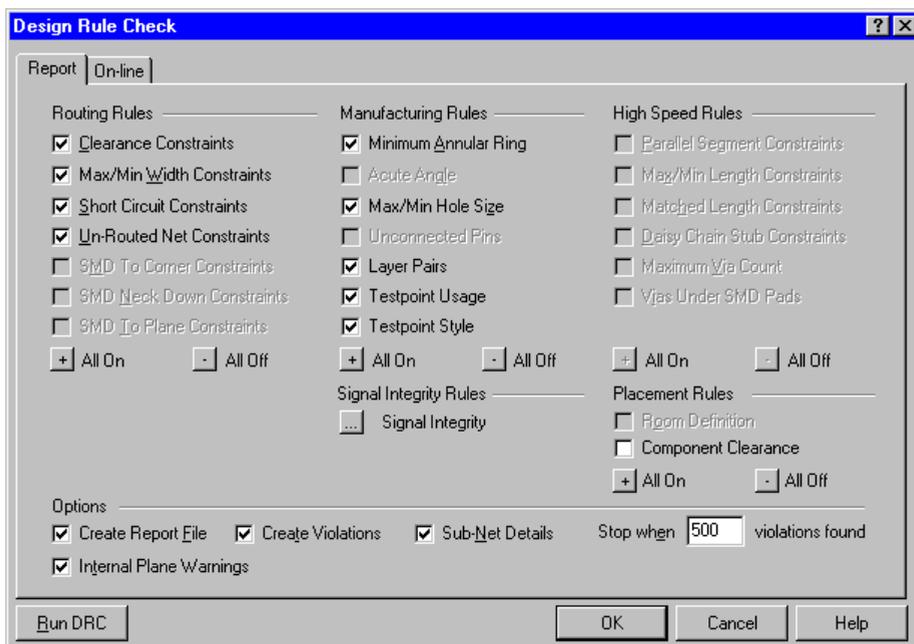


Рис. 6.68. Окно настройки пакетной проверки правил проектирования.

Выполнение проверки в пакетном режиме

После того как диалоговое окно **Design Rule Check** сконфигурировано, запуск процесса проверки производится с помощью кнопки **Run Batch**. По завершении проверки файл отчета будет автоматически открыт встроенным текстовым редактором.

Отчет программы проверки

В файле отчета программы проверки представлен список всех обработанных правил проектирования, определенных в диалоговом окне **Choose Rule Set to Check** (не включенные правила не проверяются). Каждое обнаруженное нарушение подробно описывается, то есть указывается слой, имя цепи, позиционное обозначение компонента и номер контактной площадки, а также местоположение объекта на чертеже. Нарушения зазоров, длины и ширины высвечиваются оконтуриванием объекта линией, окрашенной в специально заданный цвет **DRC**.

Устранение нарушений правил проектирования

Начинающего разработчика печатных плат подробные отчеты программы проверки могут привести в уныние. Здесь важно правильно выбрать стратегию контроля и управления нарушениями, например, ограничить число обнаруженных нарушений, установив его в поле **Stop When Found XX Violations** в диалоговом окне **Choose Rule Set to Check**. Другой стратегией является разделение процесса проверки правил проектирования на несколько этапов. Если в проекте обнаружено большое число нарушений, то следует включать по одному правилу на каждой очередной стадии проверки. Со временем каждый разработчик выработает свой собственный подход к проверке правил проектирования.

Горячая связь между файлом отчета и чертежом печатной платы



Для обработки файла отчета и исправления нарушений используется функция горячей связи (**Cross Probing**) между отчетом и чертежом печатной платы. Для этого необходимо разделить окно проекта таким образом, чтобы можно было одновременно просматривать оба документа, выбрать нужную ссылку в файле отчета и нажать кнопку **Cross Probe** на панели инструментов текстового редактора. В качестве ссылки может выступать обозначение компонента, вывод компонента, координаты X и Y или вся строка. После нажатия указанной кнопки, объект, на который указывает ссылка, появится в центре окна редактора печатных плат.

Быстрый переход к объекту на чертеже

Для быстрого перемещения к нужному объекту на плате, будь то компонент, контактная площадка, цепь, маркер ошибки, или просто в точку заданными координатами используется команда **Jump** (горячая клавиша J). При выполнении этой команды появляется меню **Jump**.

Использование панели управления для просмотра нарушений

Если на панели управления в списке **Browse** выбрать опцию **Violations**, то ниже будет выведен список всех обнаруженных нарушений. При нажатии клавишу **Details** на экран будет выведено диалоговое окно **Violations Details**, в котором отображается подробнейшая информация обо всех нарушенных правилах, а также приводятся примитивы вызывающие эти нарушения (рис. 6.69).

Поиск разорванных цепей

Разорванная цепь (цепь с одной или несколькими виртуальными линиями связи), по сути, является разомкнутой цепью. Разбиение на две подцепи (**sub-net**) означает наличие одного обрыва, на три подцепи — двух обрывов и т. д. Для нахождения разомкнутых цепей можно пользоваться следующим алгоритмом:

1. На панели управления в списке **Browse** выбирается опция **Violations**.
2. Кнопкой **Jump** изменяется масштаб вывода чертежа на экран так, чтобы отображалась вся цепь. Затем цепь подсвечивается нажатием кнопкой **Highlight**.

3. Вся цепь идентифицируется командой меню **Edit » Select » Net** (горячие клавиши **S, N**) и щелчком по любой точки цепи.
4. Подцепь идентифицируется командой меню **Edit » Select » Connected Copper** и щелчком по любому объекту цепи.

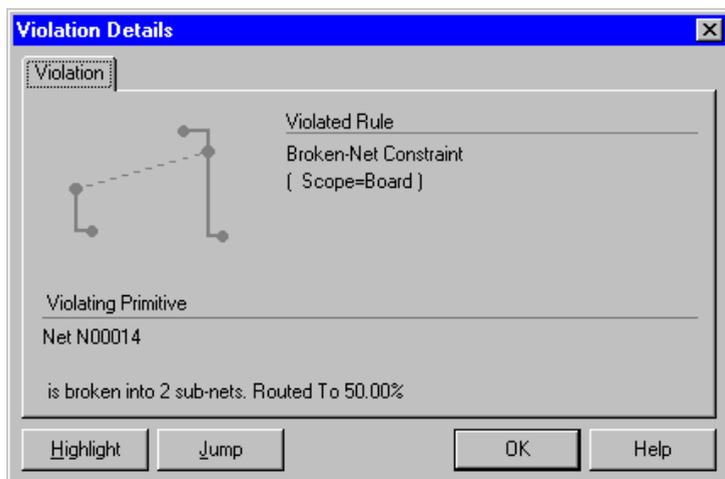


Рис. 6.69. Просмотр нарушений правил проектирования.

При выполнении команды **Select » Connected Copper** будет высвечена только несоединенная часть цепи, тогда как команда **Edit » Select » Net** подсвечивает всю цепь. Отличие этих двух команд используется для быстрого нахождения обрывов цепи, при этом необходимо в диалоговом окне **Preference** отключить опцию **Extend Selection**.

Разрешение проблемы лишних выводов

Если в отчете говорится о цепи, разбитой на множество небольших подцепей, а для другой цепи сообщается о наличии лишних выводов, принадлежащих первой, то это указывает на наличие короткого замыкания между этими цепями.

Анализ целостности сигналов

Так как сложность современных печатных плат постоянно возрастает из-за увеличения тактовых частот, повышения плотности монтажа и т. д., появилась необходимость анализа целостности сигналов в цепях платы на ранних этапах проектирования задолго до физического изготовления прототипа платы.

Система Protel 99 SE включает в себя программу контроля целостности сигналов **Simulator Integrity Simulator**, которая позволяет проанализировать компоновку печатной платы и с большой вероятностью сделать вывод о ее последующей работоспо-

собности. В качестве критериев оценки используются значения положительных и отрицательных выбросов напряжения, импеданс цепи и т. д.

В качестве исходных данных программа контроля целостности сигналов использует импедансы проводников, вычисленные специальной программой анализа линий передач, и макромодели буферов ввода-вывода. Сама программа построена на базе проверенной временем программы **Fast Reflection and Crosstalk Simulator**, обеспечивающей очень высокую точность вычислений.

Если в процессе разработки печатной платы не удастся разрешить все выявленные нарушения правил проектирования, пользователь может оценить влияние этих нарушений на функционирование платы с помощью программы анализа целостности сигналов и перекрестных искажений. При отсутствии на плате внутренних слоев питания и заземления результаты анализа могут быть менее точными.

Настройка правил проектирования для контроля целостности сигналов

Настройка правил проектирования, используемых при анализе целостности сигналов, осуществляется на вкладке **Signal Integrity** диалогового окна **Design Rules**, вызываемого командой меню **Design » Rules**.

Запуск проверки этих правил осуществляется кнопкой **Signal Integrity**, расположенной в центре диалогового окна **Design Rule Check**, вызываемого командой меню **Tools » Design Rule Check**.

◆ Для того чтобы можно было провести анализ целостности сигнала, пользователь должен включить правило **Layer Stack**.

Выполнение проверки целостности сигналов

Если заданы правила проектирования, используемые при контроле целостности сигналов, на основе которых будет производиться проверка проекта печатной платы, то после нажатия в диалоговом окне **Design Rule Check** кнопки **Run DRC** запустится программа анализа целостности сигналов (**Signal Integrity Simulator**). На экране появится небольшое окно, в котором будет показан ход процесса анализа.

После завершения процесса анализа следует настроить панель управления редактора печатных плат в режим просмотра нарушений. В этом режиме разработчик сможет изучить каждое из выявленных нарушений, щелкнув левой кнопкой мыши на кнопке **Details** для вывода подробной информации об имени цепи и действительном значении, например, выброса напряжения (рис. 6.70).

Во время такой проверки выявляется наихудший случай. То есть, каждая цепь моделируется со стороны всех возможных выходных выводов, и найденный наихудший результат проверяется на соответствие правилам проектирования.

Выполнение анализа целостности сигналов и перекрестных искажений

Если в ходе проверки правил проектирования, относящихся к целостности сигналов, были выявлены нарушения, то рекомендуется провести более подробный анализ целостности сигналов и перекрестных искажений. Вызов программы анализа осуществляется командой меню редактора печатных плат **Tools » Signal Integrity**. После чего появится окно **Protel Signal Integrity Analyzer** (рис. 6.71).

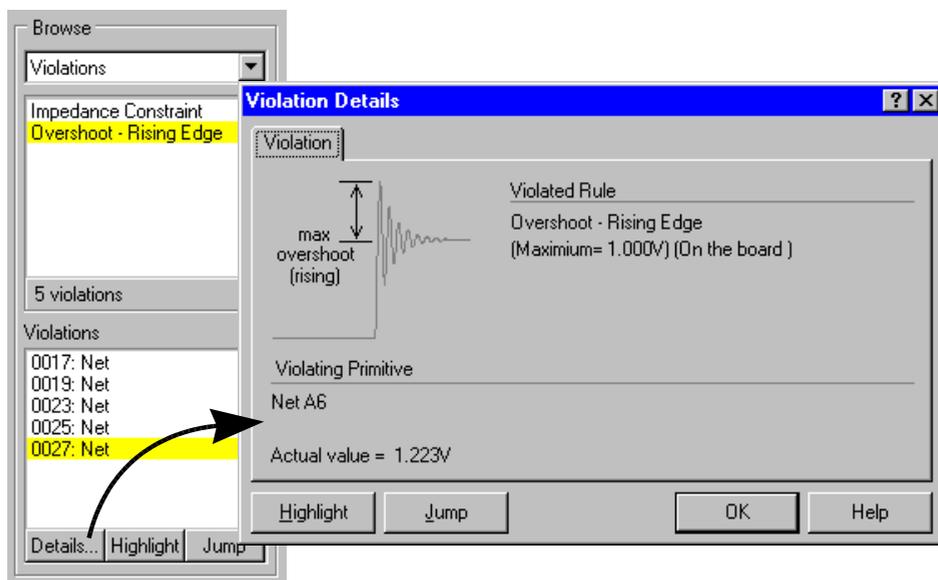


Рис. 6.70. Использование панели управления для просмотра нарушений правил проектирования при анализе целостности сигналов

Здесь в левой части окна приведен список всех цепей проекта. Для анализа цепи или нескольких цепей следует придерживаться следующего алгоритма: (цифры в кружочках на рисунке служат подсказками).

1. Необходимо выбрать цепь из списка

Для большого количества цепей проводить анализ отражений сигналов нецелесообразно. Он будет занимать много времени, а окно временных диаграмм, будет заполнено сигналами, что затруднит прочтение каких либо значимых результатов.

2. На панели инструментов нажать кнопку **Take Over Selected Net**.

Выбранная цепь появится в верхнем поле области Simulation, а список всех ее узлов будет приведен ниже. Следует заметить, что при наличии в цепи выводов, которые могут управлять схемой (двунаправленные выводы), они будут стоять первыми в списке. Чтобы изменить направление прохождения сигнала через узел, сначала необходимо его выделить, а затем нажать кнопку **In <-> Out**.

3. На панели инструментов, для запуска анализа отражений сигналов, нажать кнопку **Reflection Simulation**.

Для запуска анализа перекрестных искажений в списке цепей необходимо захватить как минимум две цепи. Затем одна из них назначается либо "агрессором", либо "жертвой". Под "агрессором" подразумевается цепь, которая вводит импульс в схему, под "жертвой" — цепь, в которой



наводятся перекрестные искажения. После того как роли распределены, нажимается кнопка **Crosstalk Simulation**.

Просмотр формы сигналов

При нажатии кнопки **Reflection Simulation** назначенные цепи анализируются, а временные диаграммы отраженных сигналов выводятся в окне **Protel WaveAnalyzer** (рис. 6.72). Пользователь может выполнить ряд измерений непосредственно по временным диаграммам. Для этого ему нужно щелкнуть мышью по узлу в списке справа от временных диаграмм и выбрать опцию в меню **Analyze**.

Если обнаружилось, что результаты, полученные по временной диаграмме, не соответствуют результатам, выданным программой проверки правил проектирования (например, программа DRC обнаружила выброс положительного импульса с амплитудой 1.2 В, а на временной диаграмме выбросы сигнала отсутствуют), это означает, что выбранный для анализа отражений сигналов выходной узел Out не является наилучшим.

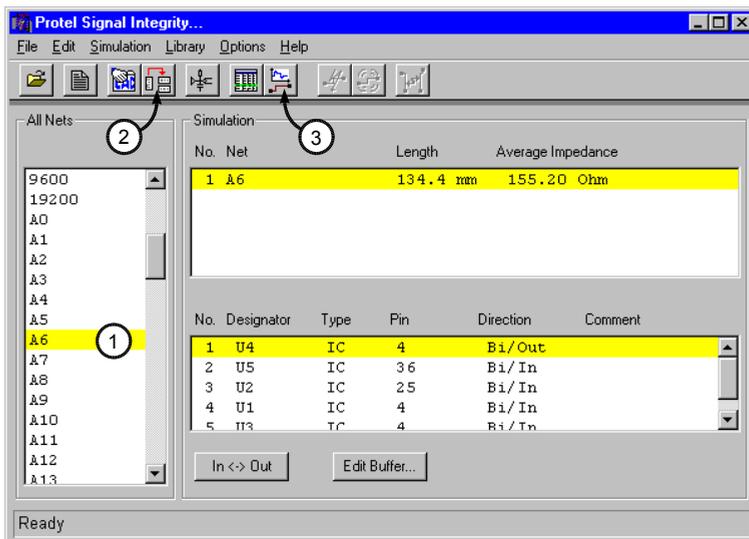


Рис. 6.71. Диалоговое окно модуля проверки целостности сигналов Protel Signal Integrity Analyzer

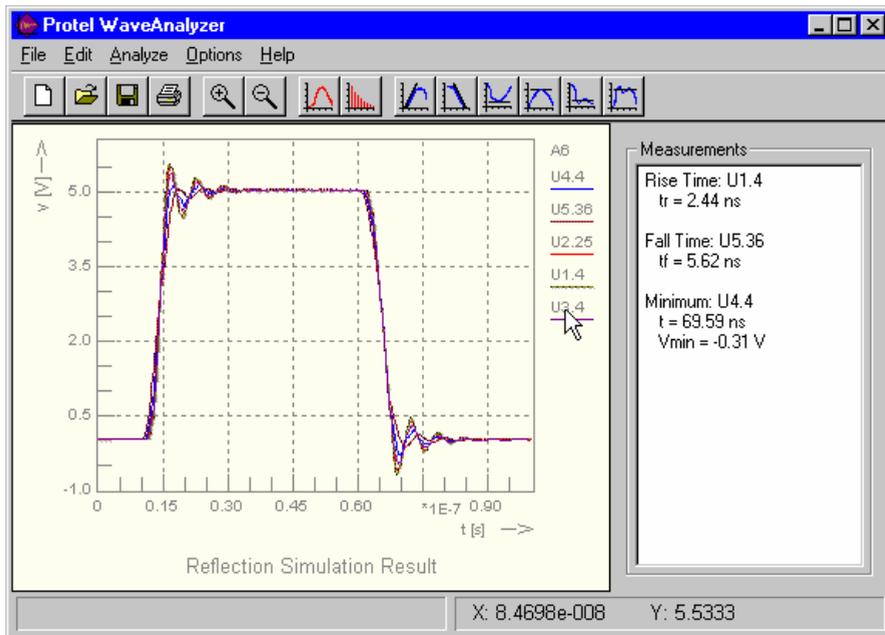


Рис. 6.72. Просмотр формы сигналов в окне Protel WaveAnalyzer.

Запуск предварительного анализа Net Screening



Кроме отражений сигналов и перекрестных искажений на платах можно оценить выбросы импульсов, задержки распространения, импедансы сегментов проводников и т. д. Результаты выводятся в виде электронной таблицы, которая цепи может быть отсортирована для быстрого поиска проблемных параметров. Для запуска анализа **Net Screening** необходимо выбрать столько цепей, сколько требуется, и нажать клавишу **Net Screening**.

Создание отчетов

Редактор печатных плат может создавать ряд отчетов, которые могут помочь разработчику при проверке состояния проекта.

Selected Pins (выделенные выводы)

Все выделенные выводы отображаются в диалоговом окне **Selected Pins**. Это удобно при проверке соединений в пределах цепи. При помощи маски (поле **Mask**) можно ограничить список выводимых на экран выводов. Файл отчета создается по нажатию кнопки **OK**.

Board Information (информация о плате)

Вкладка General

На этой вкладке содержится итоговая информация о каждом типе примитивов, используемом в проекте, размеры печатной платы (рассчитывается по самым отдаленным примитивам на чертеже печатной платы), Также здесь приведено итоговое ко-

личество контактных площадок и переходных отверстий, а также число нарушений зазоров между проводниками.

Вкладка Components

На этой вкладке приведен список всех компонентов, находящихся на верхнем и нижнем слоях, на текущий момент. Информация представлена в виде позиционного обозначения компонента и комментария к нему, если таковой имеется.

Nets Tab

Здесь содержится поименный список всех загруженных на текущий момент цепей. Информация представлена в виде ссылок на загруженные цепи.

Pwr / Gnd

Нажатием кнопки **Pwr/Gnd** вызывается диалоговое окно **Internal Plane Information**. Каждая вкладка содержит список цепей, подключенных к соответствующему внутреннему слою питания или заземления. Выбрав цепь, можно отобразить все выводы, подключенные к соответствующему слою.

Netlist Status (статус списка цепей)

Этот отчет для каждой цепи содержит список слоев, используемых при трассировке и длину окончательно разведенной цепи.

Просмотр трехмерного вида печатной платы

Система Protel 99 SE имеет возможность просмотра трехмерного вида разрабатываемой печатной платы. Здесь используется технология рендеринга OpenGL и стандартные языки, поддерживаемые подавляющим большинством графических карт. Специальный алгоритм анализирует позиционные обозначения компонентов, их топологические посадочные места и внешний контур для автоматического выбора трехмерной модели и текстур, наиболее подходящих для создания реального вида компонентов. Если данный алгоритм не может подобрать подходящей модели, выполняется обычная экструзия его контура.

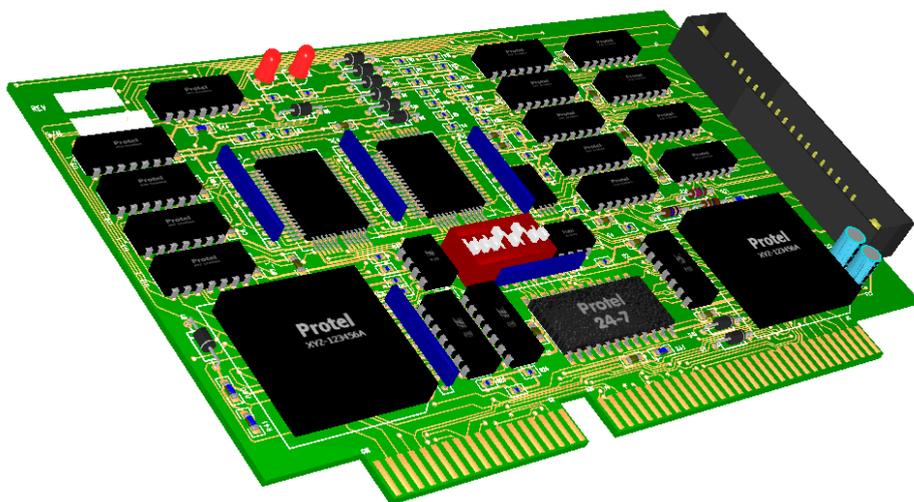


Рис. 6.73. Трехмерный вид разработанной платы.

Создание трехмерного вида платы

Для создания трехмерного вида платы используется команда меню **View » Board in 3D**. После запуска этой команды проект платы будет проанализирован и откроется окно просмотра трехмерного вида платы. Процесс рендеринга может занимать значительное время и может быть прерван щелчком левой кнопки мыши в окне просмотра трехмерного вида.

Изменение трехмерного вида платы

Модуль просмотра трехмерного вида платы позволяет выполнять его поворот и масштабирование. Поворот платы под любым углом осуществляется на панели управле-

ния модулем в окне браузера **MiniViewer** при удержании левой кнопки мыши (рис. 6.74).

В окне просмотра трехмерного вида работают некоторые стандартные горячие клавиши редактора печатных плат. Нажатие клавиш **PAGEUP** и **PAGEDOWN** позволяет изменять масштаб просмотра, нажатие клавиши **END** запускает обновление содержимого экрана, а щелчок правой кнопкой мыши вызывает инструмент **Hand** (рука), с помощью которого можно легко перемещаться по изображению при большом увеличении.

Здесь имеется возможность включать и выключать отображение на трехмерном виде компонентов, проводников, текстовых надписей и содержимого слоев для шелкографии, также используемых текстур. Выполнить это можно на панели управления или в диалоговом окне **Preferences**, вызываемом командой меню **View » Preferences**.

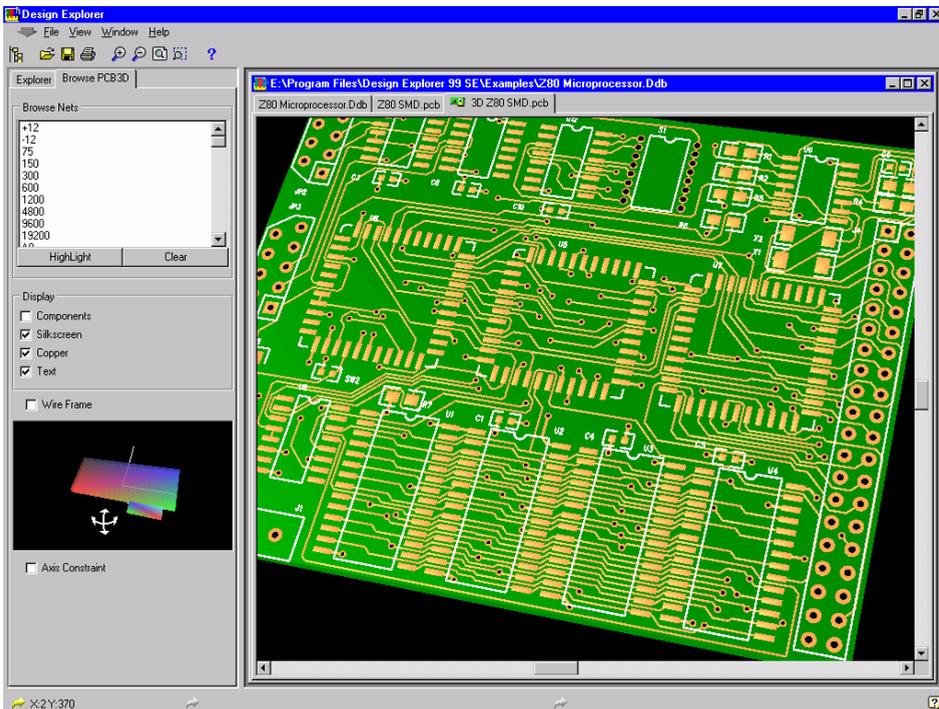


Рис. 6.74. Расположенные на панели управления опции позволяют изменять режим отображения различных объектов на трехмерном виде платы.

На панели инструментов также расположен список цепей платы. Выбрав здесь нужную цепь можно подсветить ее на трехмерном виде, нажав кнопку **Highlight**. Кнопка

Animate включает мигающий режим отображения подсвеченных цепей. Цвет подсветки задается в диалоговом окне **Preferences**.

Вывод трехмерного вида платы на печать

Распечатать трехмерный вид печатной платы можно с помощью команды меню **File » Print**.

Имеются три режима печати вида с разным качеством (**Draft**, **Normal** и **Proof**), которые устанавливаются диалоговом окне **Preferences**, вызываемом командой меню **View » Preferences**.

Создание пользовательских трехмерных моделей

Стандартный модуль просмотра трехмерного вида печатной платы использует встроенные библиотеки трехмерных моделей и не позволяет модифицировать их. Более того, эти модели являются только графическими, что означает, что они используются только для формирования изображения на экране.

Тем не менее, открытая программная архитектура системы Protel 99 SE позволяет интегрировать в среду Design Explorer дополнительные инструменты, созданные третьими фирмами.

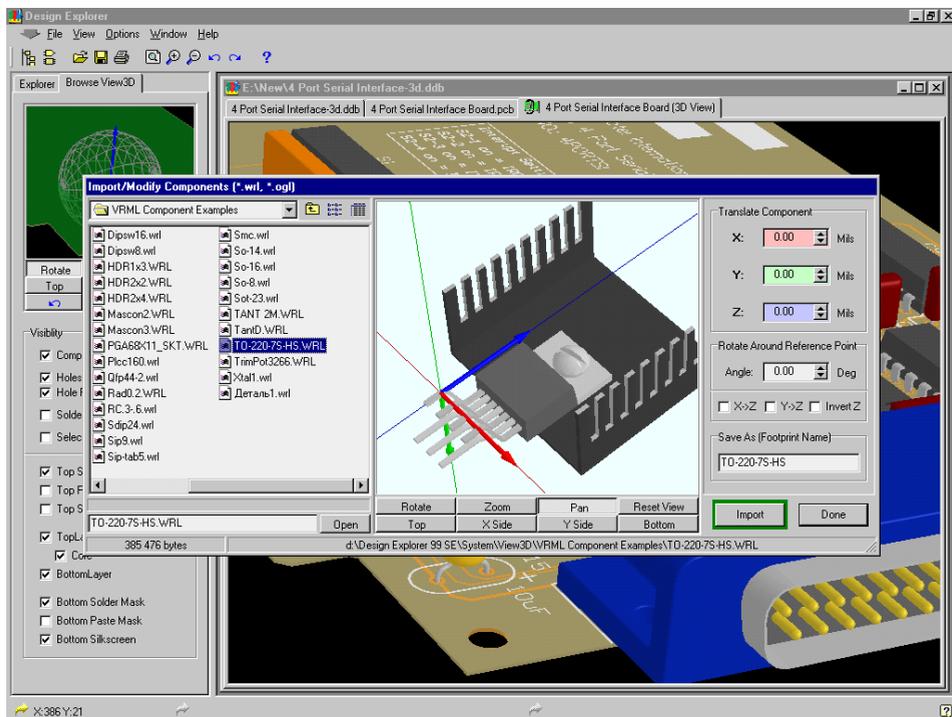


Рис. 6.75. Импорт трехмерной модели транзистора, созданной в системе SolidWorks, в Protel 99 SE с помощью специального модуля View3D компании QualECAD.

Компания QualECAD (www.qualified.com) предлагает собственный модуль просмотра трехмерного вида платы, имеющего более мощные функции управления отображением различных объектов. Здесь можно включать и выключать отдельные слои, цепи, компоненты, выводы и даже отдельные элементы стека контактной площадки. Кроме того, здесь имеется возможность импорта трехмерных моделей компонентов в формате WRL, созданных в системе SolidWorks, а также экспорт полной твердотельной модели печатной платы в эту систему.

Данный модуль добавляет в редактор печатных плат дополнительный набор правил проектирования, позволяющих на уровне DRC контролировать высоту отдельных компонентов.

Другая фирма Desktop EDA (www.desktop-eda.com.au) специализируется на разработке интерфейсов связи системы Protel 99 SE с различными пакетами твердотельного моделирования: AutoCAD, SolidWorks, ProEngineer. Механизм работы с твердотельными моделями здесь несколько иной. Модели создаются внутри механической САПР, но в Protel не импортируются. Вместо этого в проекте печатной платы задаются ссылки на соответствующие блоки, которые после экспорта проекта и загрузки в механическую САПР образуют твердотельную модель печатной платы.

Печать чертежа печатной платы средствами Windows

Основной частью процесса проектирования является выпуск документации на проект печатной платы, которая может содержать изготовительные и сборочные чертежи плат, а также контрольные чертежи слоев.

В системе Protel 99 SE вывод чертежа на печать начинается с подготовки вида для предварительного просмотра распечатки. Благодаря этому конструктор может точно задать выводимый на печать набор, правильно установить масштаб и ориентацию листа, и, что самое главное, увидеть до распечатывания как чертеж будет выглядеть на бумаге (рис. 6.76).

Модуль печати системы Protel 99 SE обеспечивает распечатку текущего содержимого экрана редактора печатных плат и копирования содержимого окна предварительного просмотра в буфер обмена системы Windows.

Функция предварительного просмотра создает документ **Power Print Configuration c**

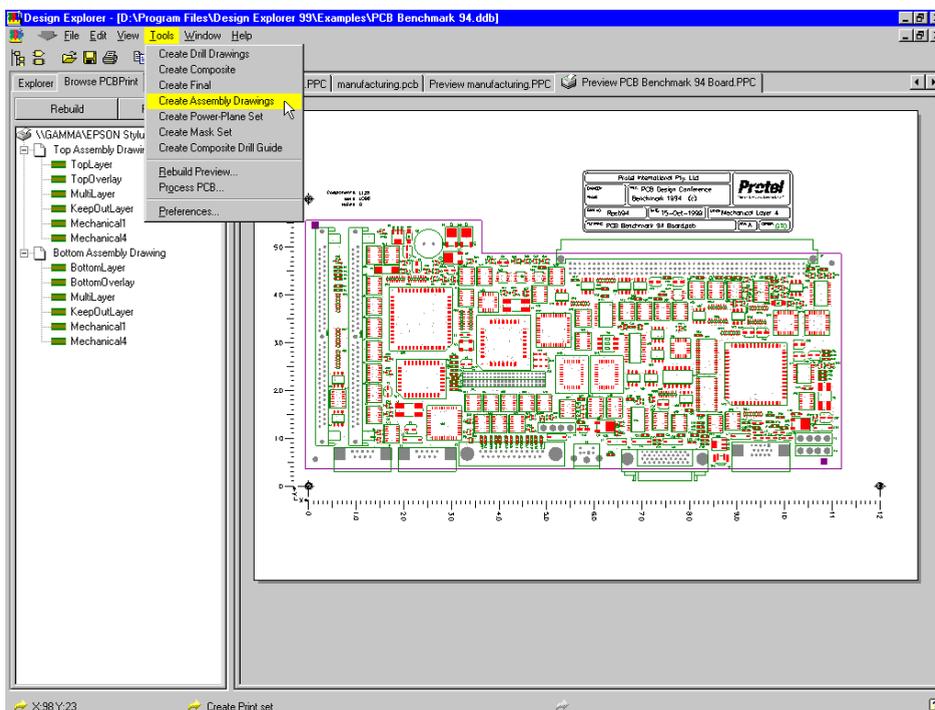


Рис. 6.76. Предварительный просмотр и конфигурация распечатки чертежа до ее вывода на печать

расширением **.PPC**. В этом документе подробно описывается, какая печатная плата находится в окне предварительного просмотра, назначенное устройство вывода, комплект распечатываемых документов и выбранные слои. При открытии **PPC** документа из него считывается вышеупомянутая информация о настройках печати, проводится анализ чертежа печатной платы, а ее изображение выводится на отдельной вкладке окна проекта для предварительного просмотра.

Так как в реальные данные проекта печатной платы в **PPC** документе не хранятся, при создании, изменении или открытии **PPC** документа они автоматически извлекаются из чертежа печатной платы. Автоматическое обновление вида предварительного просмотра можно отключить в диалоговом окне **Preferences**, после чего этот процесс будет запускаться только нажатием кнопки **Rebuild** (при изменении конфигурации предварительного просмотра) или кнопки **Process PCB** (когда изменениям подвергается печатная плата), расположенной на панели управления.

Настройки предварительного просмотра распечатки

Для вызова предварительного просмотра распечатки чертежа используется команда меню **File » Print/Preview**. После этого будет создан и открыт **PPC** документ, показывающий, как будет выглядеть чертеж печатной платы при выводе на одном или нескольких листах бумаги. **PPC** документ для проекта печатной платы **MyDesign.PCB** по умолчанию будет иметь имя **Preview MyDesign.PPC**. Отметим, что такие документы можно переименовывать аналогично любым другим документам проекта. Если файл **Preview MyDesign.PPC** уже существует, то при вызове предварительного просмотра он открывается автоматически. Для создания другой версии **PPC** документа следует изменить имя ранее созданного файла.

После того, как появится окно предварительного просмотра, текущие параметры распечатки можно изменить на вкладке **Browse PCBPrint** панели управления. По умолчанию используется текущий принтер системы Windows.

Что такое распечатка (Printout)?

Распечатка (**Printout**) представляет собой комплект, состоящий из одного или более слоев печатной платы, который будет выведен на принтер как одно задание на печать. В зависимости от выбранного масштаба она может быть разбита на один или несколько листов бумаги.

Каждая распечатка на панели **Browse PCBPrint** представлена иконкой с изображением листа бумаги (рис. 6.77).

В распечатку можно включать любую комбинацию слоев печатной платы. В одном



Рис. 6.77. Настройка параметров распечаток.

PPC документе может присутствовать практически неограниченное число распечаток.

Изменение комплекта распечаток

Вывод документов на печать в редакторе печатных плат осуществляется с помощью команды меню **File » Print**. При этом сначала выводится окно предварительного просмотра, в котором отображается комбинированная распечатка документа. Комбинированная распечатка представляет собой чертеж наложенных друг на друга основных слоев, составляющих чертеж печатной платы.

Имеется несколько заранее заданных комплектов распечаток, список которых приведен в меню **Tools**. После выбора нового набора распечаток старая конфигурация заменяется новой. Пользователь легко может добавить новую распечатку в текущий PPC документ с помощью команды меню **Edit » Insert Printout**. По умолчанию новая распечатка содержит только слой **Top Layer**.

Назначение слоев в распечатке

Чтобы раскрыть список всех слоев, назначенных на печать в данной распечатке, необходимо щелкнуть левой кнопкой мыши на значке + рядом с ее иконкой. На панели просмотра слои отражаются в том порядке, в каком они будут выведены на печать.

Чтобы добавить или удалить слой из распечатки, нужно на ее иконке выполнить щелчок правой кнопкой мыши и во всплывающем меню выбрать команду **Properties**. Появится диалоговое окно **Printout Properties**, где в списке **Layers** отображается текущий набор распечатываемых слоев (рис. 6.78). Управление слоями осуществляется с помощью кнопок, расположенных под этим списком.

Порядок действий при добавлении в распечатку нового слоя:

1. Нажатием кнопки **Add** вызвать диалоговое окно **Layer Properties**.
2. Из выпадающего списка **Print Layer Type** выбрать добавляемый слой.
3. При необходимости настроить режим отображения примитивов этого слоя. Дополнительную информацию об этом приведена в разделе *Задание стека слоев* данной главы.
4. Завершить операцию нажатием кнопки **OK**.

Новый слой появится на самой нижней позиции в списке назначенных слоев. Это означает, что при загрузке

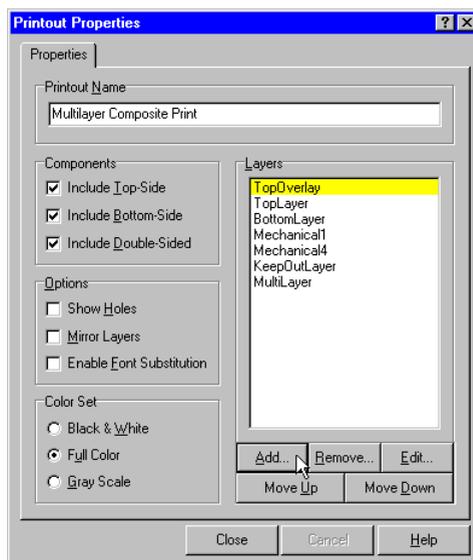


Рис. 6.78. Новые слои в распечатку добавляются в диалоговом окне **Printout Properties**.

изображения, этот слой в памяти принтера будет записан первым. Каждый вышестоящий слой записывается поверх всех нижестоящих. Порядок расположения слоев изменяется с помощью кнопок **Move Up** и **Move Down**.

Помимо описанного способа содержимое механических слоев может быть добавлено к распечатке автоматически. Для этого надо включить нужные механические слои на вкладке **Mechanical Layers** диалогового окна **Properties**.

Конфигурация печати примитивов

Конструктор может не только включать в распечатку любой слой чертежа, но и управлять тем, какие примитивы и как будут отражены в ней. Например, рассмотрим сборочный чертеж, состоящий из слоя шелкографии, верхнего слоя платы с контактными площадками для компонентов, выполненных по технологии поверхностного монтажа и слоя многослойных контактных площадок для компонентов, монтируемых в отверстия. Верхний слой будет сконфигурирован таким образом: выводится прорисовка всех примитивов компонентов (**Component Primitives**), чтобы были видны контактные площадки для поверхностного монтажа, и выключаются все остальные примитивы (**Free Primitives**), чтобы не было видно трассировки. Слой **multi-layer** конфигурируется аналогично: выводятся на печать примитивы компонентов, чтобы были видны контактные площадки со сквозными отверстиями, и выключаются остальные свободные примитивы, чтобы не было видно переходных отверстий.

Выбор принтера для печати

При создании нового **PPC** документа он автоматически настраивается на печать принтером, установленном в среде Windows по умолчанию. Можно выбрать другой принтер можно с помощью команды меню **File** » **Setup Printer**. Иначе это можно сделать, выполнив щелчок правой кнопкой мыши на иконке принтера и в появившемся всплывающем меню выбрать команду **Properties** (рис. 6.80).

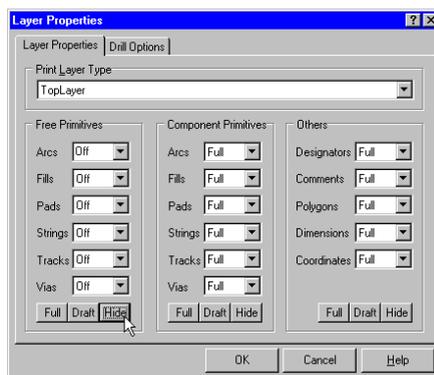


Рис. 6.79. Настройка вывода примитивов на слоях.

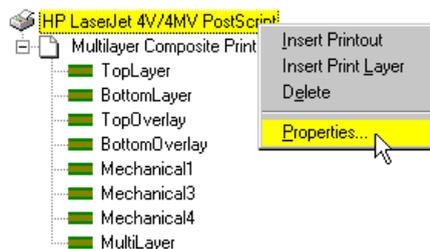


Рис. 6.80. Для изменения принтера выполните щелчок правой кнопкой мыши на иконке текущего принтера.

Настройка ориентации листа бумаги, масштаба и других параметров принтера

Ориентации листа бумаги, масштаб и поля задаются в диалоговом окне **PCB Print Options**, вызываемом командой **Properties** из всплывающего меню при щелчке правой кнопкой мыши по иконке принтера.

Имеется 3 различных заданных по умолчанию режима масштабирования. Для вывода их на экран дисплея необходимо щелкнуть по стрелке **Print What**.

Standard Print (стандартная печать) — печать чертежа печатной платы в масштабе определяемом текущим значением параметра **Print Scale**.

Whole Board on Page (вся плата на странице) — автоматически масштабирует чертеж печатной платы для печати его на одном листе бумаги.

PCB Screen Region (текущий вид на странице) — автоматически масштабирует текущий вид окна редактора печатных плат для печати его на одном листе бумаги.

Печать чертежа

После того, как все распечатки сконфигурированы в окне предварительного просмотра нужным образом, можно переходить к выводу документов на печать. В меню **File** представлено несколько режимов печати, в том числе:

Print All — вывод на принтер всех распечаток текущего **PPC** документа. Каждая распечатка посылается на печать как отдельное задание, но с одним и тем же именем.

Print Job — вывод на принтер всех распечаток текущего **PPC** документа как одно задание с именем **PPC** документа.

Print Page — вывод на принтер текущей страницы. Если распечатка разбивается на несколько страниц, они нумеруются, и модуль печати запросит у пользователя диапазон выводимых страниц. В качестве справочной информации в режиме предварительного просмотра на каждой странице многолистовой распечатки в левом верхнем углу красным шрифтом указаны номера страниц, которые на печать не выводятся. Отображением номеров страниц можно управлять опцией **Preview Page Number** в диалоговом окне **Preferences**.

Print Current — вывод на принтер всех страниц текущей распечатки.

Примечание: порядок вывода распечаток на принтер можно менять в панели просмотра, расставляя с помощью мыши распечатки в нужном порядке.

Копирование содержимого окна предварительного просмотра для работы с другими приложениями

Копирование содержимого окна предварительного просмотра чертежа печатной платы в буфер обмена системы Windows производится с помощью команды меню **Edit » Copy**. После этого скопированное изображение можно вставить в любое приложение, поддерживающее работу со стандартным буфером обмена, например, в Microsoft Word.

Примечание: По умолчанию изображение, копируется в буфер обмена как расширенный метафайл, но возможно копирование его как стандартного метафайла.

Экспорт распечаток в виде метафайлов (WMF)

Экспорт текущего набора распечаток в виде WMF файлов осуществляется с помощью команды меню **Files » Export**. В появившемся диалоговом окне **Export Options** необходимо выбрать одну из двух опций сохранения метафайла. В первом случае файл будет сохраняться один и тот же каталог с удалением старого файла, во втором — в отдельные каталоги с отметкой времени. В этой опции для каждой новой операции экспорта создается новый каталог, в имени которого указывается дата и время его создания.

Пользователь при каждом экспортировании файлов может сохранять в каталог и копии чертежа печатной платы, для чего ему потребуется включить опцию **Save a Copy of PCB File**. При этом будет сохранена копия последней сохраненной версии чертежа печатной платы.

Для экспортирования файлов непосредственно на жесткий диск следует включить опцию **Export Copy To**. Опция **Enhanced Metafile** определяет, будет ли метафайл сохранен как стандартный (16-битный) метафайл, либо как расширенный (32-битный).

Цвета, шрифты и другие установки

Цвета слоев, подмена шрифтов и другие опции задаются в диалоговом окне **Preferences**, вызываемом командой меню **Tools » Preferences**.

Вкладка Color & Gray Scale

Каждый слой в распечатке можно вывести в оттенках серого или в цвете. Оттенок серого для слоя выбирается из выпадающей палитры, появляющейся после щелчка по серому полю в таблице рядом с именем слоя. Цвет слоя выбирается в диалоговом окне **Choose Color**, появляющемся после щелчка по цветному полю здесь же.

Примечание: при выборе для каждой распечатки способа печати (в градациях серого или в цвете) следует пользоваться опциями **Set Color** в диалоговом окне **Printout Properties**.

Замена шрифтов (Font Options)

Для всех распечаток можно каждый из трех шрифтов, установленных в редакторе печатных плат поменять на любой другой шрифт системы Windows.

После того, как замена шрифтов определена и включена, необходимо открыть диалоговое окно **Printout Properties**, чтобы разрешить замену шрифтов для заданной распечатки.

Перекрытие при печати чертежа на нескольких листах (Overlap for tiled print)

Эта опция задает перекрытие изображения, выводимого на каждой странице многостраничной распечатки.

Автоматическое обновление окна просмотра

Каждый раз, когда пользователь меняет настройки окна предварительного просмотра, его содержимое обновляется. Чтобы отменить автоматическое обновление окна следует отключить опцию **Perform Automatic Rebuild for Changes**.

Если пользователь в одном из диалоговых окон изменил настройки опций, то для быстрого обновления окна предварительного просмотра следует воспользоваться командой меню **Tools » Rebuilt Preview** или нажать на панели управления кнопку **Rebuilt**. Если же пользователь модифицировал саму печатную плату, то для быстрого обновления окна предварительного просмотра следует воспользоваться командой меню **Tools » Process PCB** или нажать на панели управления кнопку **Process PCB**.

Вкладка Mechanical Layers

Здесь задается режим добавления содержимого механических слоев во все создаваемые распечатки, а не в отдельные из них.

Генерация выходных файлов для производства

Разработка печатной платы является всего лишь первым этапом процесса, кульминацией которого является производство и изготовление печатной платы. Связующим звеном между проектом и готовой печатной платой является ее чертеж, создаваемый с помощью функций вывода на печать или генерации файлов в формате Gerber и таблиц сверления NC Drill, а также список используемых материалов (Bill of Materials), рапорт об использовании контрольных точек и файл для оборудования размещения.

Информация о выводе чертежа на фотоплоттеры приведена в подразделе *Какие бывают фотоматрицы?* данной главы. Печать чертежей с помощью стандартных устройств описана в разделе *Печать чертежа печатной платы средствами Windows*.

Взаимодействие с производителями печатных плат

Перед началом проектирования необходимо ясно представлять себе требования, накладываемые на его конечный результат, и учитывать особенности технологического цикла производства печатных плат.

Если для изготовления фотоматриц привлекаются сторонние организации, специализирующиеся исключительно на этом, разработчик должен соблюдать все заранее оговоренные технологические требования. Например, в некоторых случаях может потребоваться нанесение на чертеж специальных реперных символов или выполнение панелизации.

При проектировании рекомендуется знать максимально допустимый для фотоплоттера размер пленки, требования на зазоры, а также предполагаемые производственные допуски. Подобные же соображения требуются при планировании сверления отверстий на станках с числовым программным управлением.

В некоторых случаях предпочтительно, чтобы оборудование работало непосредственно с "сырыми" Gerber файлами (или даже PCB файлами), а не с панелизованными файлами. Понимание этих требований поможет правильно спланировать процесс проектирования и выполнить разработку эффективно и без особых проблем.

Генерация выходных файлов с помощью модуля CAM Manager

Модуль **CAM Manager** представляет собой независимый редактор, работающий в среде Design Explorer совместно с другими редакторами различных частей проекта и осуществляющий полное управление процессом создания выходных файлов для производства, включая файлы в формате Gerber, файлы для сверлильных станков с ЧПУ, файлы для оборудования размещения компонентов, перечней используемых материалов, отчетов о добавлении контрольных точек и о проверке правил проектирования. Все настройки сохраняются в проекте в специальном документе **CAM Output**

Configuration с расширением **.CAM** и могут быть изменены в любое время. При необходимости этот документ может быть легко скопирован в другие проекты, чем облегчается перенос конфигурации.

В состав модуля **CAM Manager** входит специальный мастер **CAM Output Wizard**, облегчающий пользователю настройку и создание выходных файлов.

Все доступные выходные файлы могут быть сгенерированы в любой момент работы над проектом с помощью всего одной команды. Генерируемые файлы записываются в проект в отдельную папку **CAM Outputs**, к имени которой может быть добавлена информация о времени и дате создания. Имеется возможность автоматического экспорта выходных файлов непосредственно на диск компьютера.

С помощью редактора **CAM Manager** могут быть созданы выходные файлы следующих типов:

- Файлы в формате Gerber;
- Файлы для сверления отверстий на станках с числовым программным управлением (NC Drill);
- Отчеты о добавлении контрольных точек;
- Управляющие файлы для оборудования автоматического позиционирования компонентов (Pick and Place);
- Перечни используемых материалов (Bill of Materials, BOM);
- Отчеты программы проверки правил проектирования (Design Rule Check, DRC);

Создание нового CAM документа

Для создания нового **CAM** документа в текущем проекте необходимо выполнить команду меню **File** » **CAM Manager**, после чего будет создан пустой **CAM** документ и автоматически запущен мастер **CAM Output Wizard**, с помощью которого задаются настройки для выходных файлов всех поддерживаемых типов. Изменение настроек можно производить напрямую, выбрав нужный тип файлов и выполнив команду меню **Edit**.

Присвоение имени CAM документу

После запуска команды **File** » **CAM Manager** файлу **CAM** документа автоматически присваивается имя **CAM Outputs for MyDesign.CAM** для чертежа печатной платы с именем **MyDesign.PCB**. Если **CAM** документ с таким именем уже существует, он просто открывается. Если пользователю нужно создать второй **CAM** документ для данной платы, перед запуском модуля **CAM Manager** необходимо переименовать существующий файл.

Добавление настроек выходных файлов в CAM документ

Существует два способа добавления новых настроек выходных файлов в **CAM** документ: с помощью специального мастера, либо с помощью соответствующей команды меню **Edit**.

Использование мастера CAM Output Wizard

Мастер может быть запущен с помощью команды меню **Tools** » **CAM Wizard** или посредством щелчка правой кнопкой мыши по **CAM** документу и последующему выбору команды **CAM Wizard** в появившемся контекстном меню. С помощью этого мастера можно выполнить настройку любого типа выходных файлов, для чего необходимо выбрать требуемый тип во втором окне мастера. Если при прохождении через управляющие окна мастера пользователь не уверен в том, какое значение нужно установить в том или ином поле, рекомендуется оставить значение по умолчанию. Любые настройки могут быть изменены позднее посредством двойного щелчка по нужной настройке в **CAM** документе, после чего появится соответствующее диалоговое окно. Для получения подробной информации о какой-либо конкретной опции можно воспользоваться иконкой помощи **What's This?**, расположенной в правом верхнем углу окна.

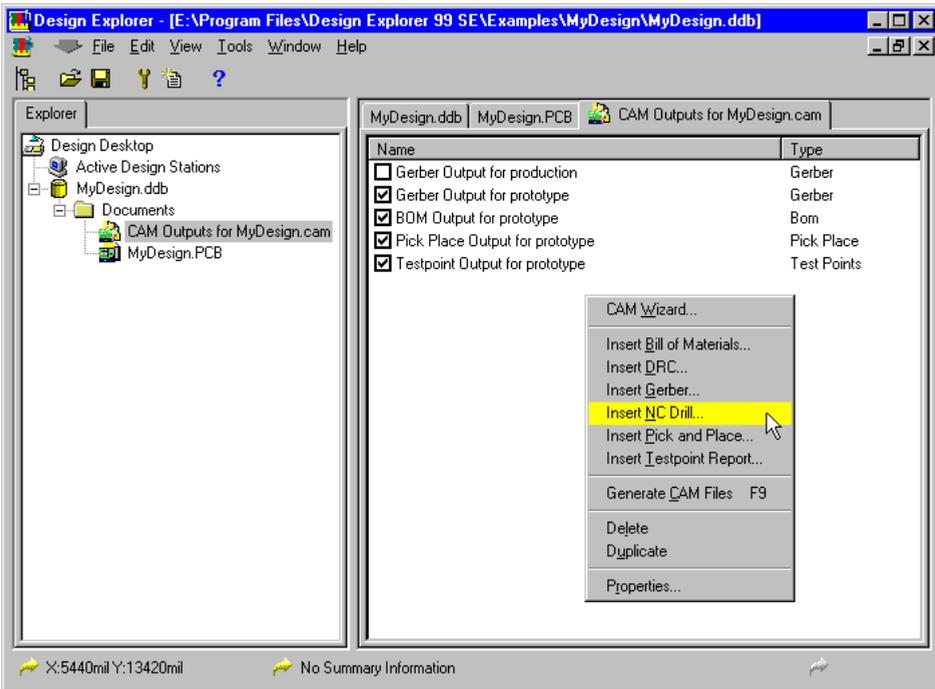


Рис. 6.81. Для добавления новой настройки в CAM документ выполните щелчок правой кнопкой мыши в окне модуля CAM Manager.

Прямое добавление настроек выходного файла

Любой **CAM** документ может хранить несколько настроек, включая настройки одного типа. Например, некоторый **CAM** документ может содержать **Gerber** и **NC drill** настройки для изготовления макета, а также второй набор настроек для выпуска серийного изделия.

Для добавления новой настройки необходимо выбрать требуемый тип выходного файла в командном меню **Edit** модуля **CAM Manager** или в контекстном меню, вызываемом нажатием правой кнопки мыши (рис. 6.81). Появится диалоговое окно **Setup** для этого типа, где выбирается необходимая конфигурация, после чего для закрытия окна нужно нажать кнопку **OK**.

Изменение, копирование и удаление настроек выходных файлов

Для изменения существующих настроек выходного файла необходимо выполнить на нем двойной щелчок левой кнопкой мыши. Для удаления или копирования используются соответствующие команды из контекстного меню, вызываемого нажатием правой кнопки мыши.

Конфигурация опций процесса генерации выходных файлов

Опции процесса генерации выходных файлов конфигурируются в диалоговом окне **CAM Options**, вызываемом командой меню **Tools** » **Preferences**. Ниже приводится подробное их описание.

Поле **CAM Output Folder** (папка для выходных файлов)

Выходные файлы записываются в новую папку, создаваемую в месте, что хранится файл печатной платы. Если включена опция **Overwrite folder**, то эта папка получит название **CAM for MyDesign** (для чертежа печатной платы **MyDesign.PCB**), а все выходные файлы будут записаны в нее после запуска команды меню **Generate CAM Files**.

Если включена опция **Create Time-Stamped Output Folder**, то каждый раз при запуске команды **Generate CAM Files** будет создаваться новая папка, к имени которой будут добавлены текущие дата и время.

Поле **CAM Output Files Destination** (местоположение выходных файлов)

Если включена опция **One Folder for All Outputs**, все выходные файлы записываются в главную **CAM** папку. Если же включена опция **Separate Folder for Each Output Type**, в ней создаются подпапки, соответствующие каждому типу выходных файлов.

Поле **Archive PCB File Option** (сохранение архивной копии чертежа платы)

Если опция **Save a Copy of the PCB** включена, при генерации выходных файлов последняя сохраненная версия чертежа печатной платы копируется в папку выходных файлов.

Поле **Export CAM Outputs** (экспорт выходных файлов)

Включение опции **Export Copy To** позволяет автоматически выполнять экспорт выходных файлов с сохранением их непосредственно на диск компьютера в назначенную папку.

Генерация выходных файлов

Выходные файлы могут быть сгенерированы из существующих настроек в любой момент работы над проектом. Для генерации того или иного выходного файла необходимо включить (поставить "галочку") соответствующую строку настроек. После этого выполняется команда меню **Tools » Generate CAM Files** (горячая клавиша **F9**), по которой создаются все выбранные на предыдущем шаге выходные файлы согласно текущим настройкам, установленным в диалоговом окне **CAM Options**.

Генерация выходных файлов в формате Gerber

Что представляют собой файлы в формате Gerber?

Печатная плата представляет собой набор слоев, которые производитель компоует в единое целое посредством выполнения последовательности различных химических и механических технологических операций. Для изготовления какого-либо физического слоя печатной платы используется фотошаблон, представляющий собой кусок прозрачной пленки, содержащий рисунок проводников данного слоя, созданный редактором печатных плат.

После завершения процесса разработки печатной платы и выполнения всех проверок правил проектирования Gerber файлы генерируются по одному на каждый задействованный в производственном процессе слой. Язык Gerber представляет собой стандартное средство, используемое для передачи данных из системы разработки печатных плат в систему изготовления фотошаблонов. Сгенерированные файлы в формате Gerber передаются производителю, который загружает их в фотоплоттер и изготавливает фотошаблоны.

Фотошаблон для некоторого слоя создается на пленке путем засвечивания светочувствительной эмульсии пятном света, повторяющим форму графических примитивов на чертеже этого слоя. То есть, фотоплоттеру нужна информация о форме пятна света и координаты его перемещения. Различные формы пятна света задаются апертуррами, которые, как правило, синтезируются из чертежа платы и включаются в Gerber файлы. В этом случае апертюры называются внедренными. В иных случаях апертюры сохраняются в отдельном файле, который передается производителю вместе с набором Gerber файлов.

Настройка генерации файлов в формате Gerber

Для включения настройки генерации файлов в формате Gerber необходимо выполнить команду меню **Edit » Insert Gerber**, после чего появится диалоговое окно **Gerber Setup**. Для получения подробной информации о любой из опций данного окна можно воспользоваться иконкой **What's This?**, расположенной в правом верхнем углу окна.

Заметим, что при включении опции **Embedded Apertures** (внедренные апертюры), расположенной на вкладке **Apertures** диалогового окна **Gerber Setup**, список апертюр создается автоматически каждый раз при генерации Gerber файлов и добавляется в эти файлы согласно стандарту RS274X. Эта возможность устраняет необходимость контроля за полнотой списка апертюр, поэтому рекомендуется всегда включать данную опцию, кроме случаев, когда изготовитель печатных плат не поддерживает обработку внедренных апертюр. Более подробная информация по этому вопросу приведена в разделе *Какие бывают фотошаблоны?* данной главы.

Для того чтобы убедиться, что законченная печатная плата соответствует проекту и удовлетворяет требованиям производителя, перед генерацией Gerber файлов необходимо обсудить с производителем следующие вопросы:

Ограничения на апертюры – Большинство современных фотоплоттеров являются растровыми и поддерживают апертюры любого размера, а также допускают применение Gerber файлов с внедренными апертюрами. В этом случае рекомендуется включить опцию **Embedded Apertures**, расположенную на вкладке **Apertures** диалогового окна **Gerber Setup**.

Отступы трафаретов – Отступы трафаретов нужны для генерации масок для нанесения паяльной пасты или пайки волной. Окна в таких трафаретах определяют места на металлизированных участках платы, где должен быть нанесен припой (обычно это контактные площадки для монтажа компонентов) или нанесена паяльная паста (для компонентов поверхностного монтажа). Все эти параметры определяются соответствующими правилами проектирования, задаваемыми на вкладке **Manufacturing** диалогового окна **Design Rules** редактора печатных плат.

Зазоры на внутренних слоях питания – Если в проекте присутствуют внутренние слои питания, производитель должен определить зазоры, которые он может реализовать для изоляции сквозных контактных площадок и переходных отверстий, не соединяемых с этими слоями. В проекте зазоры устанавливаются с помощью соответствующих правил проектирования, задаваемых на вкладке **Manufacturing** диалогового окна **Design Rules** редактора печатных плат.

Система единиц измерения и формат записи чисел – Необходимо выяснить, в какой системе единиц должны быть представлены координаты примитивов: в дюймовой или метрической. Формат записи чисел определяет точность размещения примитивов на фотошаблоне. Например, формат **2:3** имеет разрешение 1 мил (тысячная доля дюйма). Если проект содержит объекты, размещенные в сетке с шагом меньшим 1 мил, то формат **2:3** для него не подойдет. С другой стороны, необоснованное завышение точности может сильно усложнить процесс изготовления фотошаблона и увеличить его стоимость.

Центровка фотошаблона на пленке – Центровка фотошаблона на пленке заданного размера может быть выполнена автоматически при включении опции **Center Plots On Film** в диалоговом окне **Gerber Setup**. Заметим, что если эта опция выключена, координаты отсчитываются от точки абсолютного начала координат.

Требования на сверление отверстий – данные о параметрах сверления отверстий обычно передаются на производство в виде специальных файлов для сверлильных станков с ЧПУ. Более полная информация по этому вопросу приведена в подразделе *Генерация файлов для сверления отверстий (NC Drill)*.

Идентификация Gerber файлов с помощью расширения

При генерации выходных файлов в формате Gerber создается набор файлов, каждый из которых соответствует указанному в настройках слою. В последствии эти файлы загружаются в фотоплоттер, где по ним изготавливаются отдельные фотошаблоны.

К именам Gerber файлов расширение добавляется автоматически, оно является уникальным и определяет слой и тип печати. Например, файл печати верхнего слоя платы, имеющей имя **MyDeisgn**, будет сохранен как **MyDeisgn.GTL**, где расширение **.GTL** обозначает **Gerber-Top Layer** (верхний Gerber слой). Поскольку в каждом проекте может быть сгенерировано несколько выходных файлов, такая система расширений помогает идентифицировать комплекты выходных файлов.

Рекомендуется использовать общепринятую систему присвоения расширений файлов. В таблице 6.1 перечислены расширения Gerber файлов, принятые в системе Protel 99 SE.

Таблица 6.1

Слой	Расширение
Top Overlay	.GTO
Bottom Overlay	.GBO
Top Layer	.GTL
Bottom Layer	.GBL
Mid Layer 1, и т.д.	.G1, .G2, и т.д.
Power Plane 1, и т.д.	.GP1, GP2, и т.д.
Mechanical Layer 1, и т.д.	.GM1, .GM2, и т.д.
Top Solder Mask	.GTS
Bottom Solder Mask	.GBS
Top Paste Mask	.GTP
Bottom Paste Mask	.GBP
Drill Drawing	.GDD
Drill Drawing – Top to Mid 1, Mid2 to Mid 3, и т.д.	.GD1, GD2, GD3, и т.д.
Drill Guide	.GDG
Drill Guide – Top to Mid 1, Mid 2 to Mid 3, и т.д.	.GG1, GG2, GG3, и т.д.
Pad Master, Top	.GPT
Pad Master, Bottom	.GPB
Keep Out Layer	.GKO
Gerber Panels	.P01, .P02, и т.д.

Генерация файлов для сверления отверстий (NC Drill)

Требования на файлы сверления отверстий на станках с числовым программным управлением конфигурируются в диалоговом окне **NC Drill Setup**, которое появляется при выполнении команды меню **Edit » Insert NC Drill** или двойном щелчке на соответствующей строке настроек в **CAM** документе.

Эти файлы используются для автоматического программирования станков для сверления отверстий в процессе производства печатной платы. В данных файлах определяются размеры и координаты, а также код инструмента. Координаты отверстий отсчитываются от заданного пользователем относительного начала координат.

Генерируется три типа файлов для сверления:

МуPCB.DRR - отчет, содержащий список файлов, сгенерированных для каждой пары слоев. Здесь содержатся используемые инструменты, размеры отверстий в метрических и дюймовых единицах измерения, количество отверстий и карта перемещения инструмента.

МуPCB.DRL - Файл формата Excellon, имеющий расширение **.DRL**. Если проект включает несколько слоев и используются глухие или скрытые переходные отверстия, то для каждого технологического слоя будут также сгенерированы отдельные файлы, которым будут присвоены расширения **.DR1**, **.DR2** и т. д.

МуPCB.TXT - Для каждого файла формата Excellon будут созданы его ASCII версия с расширениями **.TXT**, **.TX1**, **.TX2** и т. д.

Настройка генерации файлов сверления (NC Drill)

Файлы для сверления отверстий должны быть созданы с такой же точностью (то есть, иметь такой же формат), что файлы в формате Gerber. Например, если выходной файл в формате Gerber был сконфигурирован с форматом чисел **2:4**, то соответствующий ему выходной файл сверления должен иметь такой же формат.

Генерация перечней используемых материалов (Bill of Materials)

Требования на файлы, содержащие перечни используемых материалов (Bill of Materials, BOM), конфигурируются в диалоговом окне **Bill of Materials Setup** (рис. 6.82), появляющемся после выполнения команды меню **Edit » Insert Bill of Materials** или двойном щелчке левой кнопкой мыши соответствующей строке **CAM** документа.

Перечни могут быть сгенерированы в любом из трех форматов BOM файлов: текстовом, **CSV (Comma Separate Value** — значения, разделенные запятой) и электронной таблицы (**Spreadsheet**). Текстовый и **CSV** форматы группируют перечень материалов по комментариям к компонентам и значению. Вариант электронной таблицы является более гибким и позволяет пользователю задавать формат перечня материалов.

Настройка генерации перечней используемых материалов (Bill of Materials)

От пользователя требуется выбрать нужный выходной формат перечня, причем допускается генерация более одного выходного файла. В этом случае они будут иметь разные расширения, что облегчит их идентификацию.

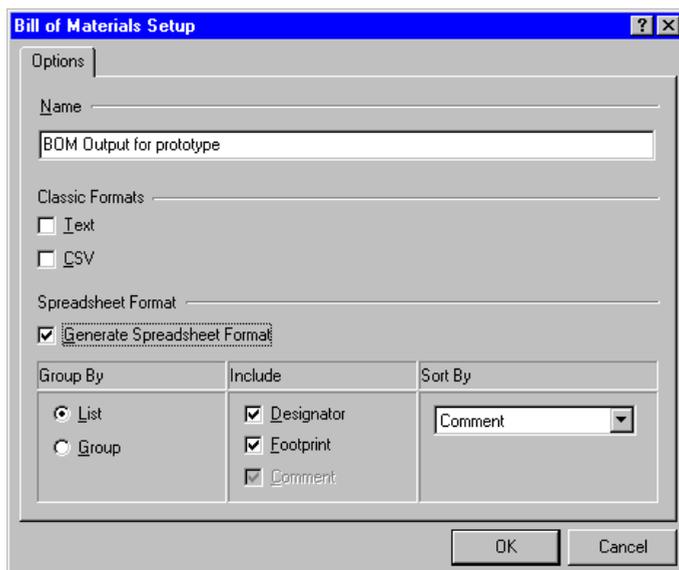


Рис. 6.82. Генерировать перечни используемых материалов можно в любом из трех форматов.

Генерация файлов для оборудования размещения (Pick and Place)

Требования на файлы, управляющие автоматическим оборудованием размещения компонентов на плате (**Pick and Place**), конфигурируются в диалоговом окне **Pick and Place Setup**, появляющемся после выполнения команды меню **Edit » Insert Pick and Place** или двойном щелчке левой кнопкой мыши соответствующей строке **CAM** документа.

Такие файлы содержат следующую информацию о каждом из компонентов:

- Позиционное обозначение;
- Тип топологического посадочного места;
- Местоположение на плате — может задаваться тремя способами: координатами геометрического центра компонента, координатами его точки привязки и координатами контактной площадки с номером 1.

- Сторона платы;
- Угол поворота компонента;
- Текстовый комментарий к компоненту.

Файлов для оборудования размещения (**Pick and Place**) могут быть сгенерированы в любом из трех форматов: текстовом, **CSV (Comma Separate Value** — значения, разделенные запятой) и электронной таблицы (**Spreadsheet**). Координаты отсчитываются от заданного пользователем относительного начала координат.

Настройка генерации файлов для оборудования размещения

От пользователя требуется выбрать нужный выходной формат файла, причем допускается генерация более одного выходного файла. В этом случае они будут иметь разные расширения, что облегчит их идентификацию.

Генерация отчета о контрольных точках (Testpoint Report)

Требования на файлы, содержащие отчеты о добавленных контрольных точках, конфигурируются в диалоговом окне **Testpoint Setup**, появляющемся после выполнения команды меню **Edit » Insert Testpoint** или двойным щелчком левой кнопкой мыши соответствующей строке **CAM** документа.

Контрольные точки предназначены для подключения к плате тестирующего оборудования. Тестирование печатных плат бывает двух видов: проверка правильности изготовления до сборки и проверка функционирования всей платы после сборки. В первом случае используется специальное контактное приспособление, во втором случае подключаются щупы измерительного оборудования.

На платах контрольные точки являются не чем иным, как контактными площадками или переходными отверстиями, у которых включен атрибут **Testpoint** и которые могут располагаться на нижней, верхней или обеих сторонах платы.

Контрольные точки могут быть определены вручную с помощью функции поиска **Testpoint Find** или размещены программой автоматической трассировки. Их размещение управляется специальными правилами проектирования **Testpoint Style** (стиль контрольных точек) и **Testpoint Usage** (использование контрольных точек). Отчет о проверке правил **Testpoint Style** содержит те контрольные точки, в которых не соблюдены требуемые физические параметры (размеры, зазоры и т. п.), а отчет о проверке правила **Testpoint Usage** включает те цепи, к которым назначенные контрольные точки не были добавлены по каким-либо причинам. Более подробная информация по этому вопросу приведена в разделе *Определение требований к проекту печатной платы*.

Отчет о контрольных точках затем используется для поиска всех контактных площадок и переходных отверстий, у которых включен один из атрибутов контрольной точки.

- Имя цепи;
- Имя контрольной точки;
- Координаты X и Y — отсчитываются от заданного пользователем относительного начала координат;
- Сторона платы;

- Диаметр отверстия;
- Тип контрольной точки — сквозная или поверхностная;

Настройка генерации отчета о контрольных точках

В окне настроек выбирается выходной формат отчета. Существует возможность выбора четырех форматов, при этом будут созданы файлы с разными расширениями.

Генерация отчета о проверке правил проектирования (Design Rule Check)

Требования на файлы, содержащие отчеты о проверке правил проектирования (**Design Rule Check**), конфигурируются в диалоговом окне **DRC Setup**, появляющемся после выполнения команды меню **Edit » Insert DRC** или двойным щелчком левой кнопкой мыши соответствующей строке **CAM** документа.

Проверка правил проектирования и создание соответствующего отчета могут быть выполнены непосредственно в редакторе чертежей печатных плат с помощью команды меню **Tools » Design Rule Check**. Модуль **CAM Manager** запускает те же самые процедуры проверки, так как они облегчают процесс подготовки необходимых выходных файлов, когда этап проектирования завершен и проект готов к передаче на производство. Подробную информацию о проверке правил проектирования можно найти в разделе *Верификация проекта печатной платы*.

Настройка генерации отчета о проверке правил проектирования

В окне настроек **DRC Setup** включаются все установленные в проекте правила, которые требуют проверки.

Перенос CAM документа в другой проект

CAM документ может быть скопирован из одного проекта в другой подобно любому другому документу. Перед копированием необходимо закрыть **CAM** документ, щелкнуть правой кнопкой мыши по его иконке на дереве проекта. Появится контекстное меню, в котором нужно выбрать команду **Copy**. Далее необходимо щелкнуть правой кнопкой мыши в папке другого проекта, куда будет копироваться документ, и в появившемся меню выбрать команду **Paste**. Также для копирования из одного проекта в другой можно использовать перетаскивание подобно тому, как это делается в Проводнике системы Windows при копировании файлов.

При копировании **CAM** документа, как правило, потребуется заново задать исходный файл печатной платы. Это может быть сделано в любое время работы с документом в модуле **CAM Manager** с помощью команды **Tools » Set Target Board**.

Замечание: Если пользователь заново задает имя исходной платы и не использует опцию внедренных апертур (**Embedded Apertures**) для Gerber файлов, то перед генерацией Gerber файлов необходимо убедиться, что список апертур соответствует примитивам на новой плате.

Какие бывают фотошаблоны?

Фотошаблоном называется лист светочувствительной пленки, на которую в процессе изготовления был нанесен рисунок, повторяющий расположение графических примитивов на соответствующем слое печатной платы. Как правило, фотошаблоны изготавливаются на фотоплоттерах, принимающих данные в формате Gerber.

Фотошаблоны также могут быть изготовлены с помощью фотонаборных устройств высокого разрешения, получающие входные данные в формате Postscript. Это оборудование способно производить пленочные позитивы с разрешением 2540 точек на дюйм и может стать достаточно дешевой альтернативой выводу в формате Gerber.

Однако, пользователи должны хорошо представлять себе ограничения этого способа печати чертежей печатных плат. Высокое разрешение этих систем вовсе необязательно означает высокую точность позиционирования или линейность, особенно при печати больших чертежей. Выходные файлы в формате Postscript генерируются с помощью функции **Power Print** редактора печатных плат.

Изготовление фотошаблонов на фотоплоттерах

Печать чертежей на фотоплоттерах, использующих формат Gerber, обеспечивает наибольшее разрешение и по праву считается лучшим способом подготовки печатных плат к производству, так как обеспечивает наилучшее качество фотошаблона. Использование фотоплоттеров является абсолютно необходимым, если требуется обеспечить высокую линейную точность изготовления платы большого размера с высокой плотностью размещения компонентов. Файлы в формате Gerber генерируются специальным модулем **CAM Manager** редактора печатных плат.

Общие сведения о фотоплоттерах

Фотоплоттеры во многом похожи на перьевые плоттеры, но печать осуществляется лучом света непосредственно на светочувствительной пленке. Многочисленные преимущества такого способа привели к широкому распространению фотоплоттеров в электронной промышленности.

Так как травление печатных плат, как правило, базируется на фотографической технологии, производство позитивных и негативных фотошаблонов является неотъемлемой частью этого процесса. Если печать чертежа выполняется на перьевых плоттерах, то для получения конечного фотошаблона необходимо выполнить некоторое количество промежуточных шагов. В этом случае для достижения приемлемой точности печать обычно выполняется в масштабе не менее 2:1, после чего изображение уменьшается фотографическим способом.

Фотоплоттеры же обеспечивают достаточную точность для печати даже в масштабе 1:1. Имеется ряд организаций, специализирующихся на изготовлении фотошаблонов по вполне приемлемым ценам, которые могут помочь разработчикам в решении этой проблемы. Для того чтобы сделать правильный выбор, необходимо понимание некоторых ключевых концепций фотопечати.

Сравнение векторных и растровых плоттеров

Фотоплоттеры делятся на две основные категории — векторные и растровые.

Векторные плоттеры строят изображение из комбинации "мазков" и "точек", формируемых световым пятном. Изображение в этом случае похоже на то, что прорисовывается перьевыми плоттерами, которые выбирают инструмент нужной толщины и описывают в пространстве чертежа некоторый вектор, после чего на нем образуется линия, ширина которой определяется диаметром апертуры. Набор апертур позволяет плоттеру прорисовывать проводники различной ширины, контактные площадки различной формы и многое другое. Некоторые плоттеры используют для линий и точек разные апертуры, другие управляют интенсивностью света, в этом случае любая из апертур может быть использована как для линий, так и для точек.

Растровые плоттеры не используют систему фиксированных апертур. Они читают Gerber файл, хранящий изображение всего чертежа, формируют его растровое изображение и выводят на пленку построчно, аналогично тому, как формируется изображение на экране телевизора. Растровые фотоплоттеры могут синтезировать практически неограниченное число различных апертур, являются более гибким инструментом и предоставляют разработчикам дополнительные возможности.

Некоторые фотоплоттеры используют формат Postscript. Файлы для таких устройств создаются с применением соответствующих драйверов Postscript. Более подробная информация по этому вопросу приведена в пункте *Советы по Postscript печати* данной главы.

В общем случае для большей эффективности проектирования рекомендуется учитывать тип фотоплоттера, на котором будет изготавливаться конечный фотошаблон платы.

- ◆ Перед окончательной генерацией файлов для печати свяжитесь с организацией, которая непосредственно будет изготавливать фотошаблон. Правильный выбор опций печати на этом этапе позволит значительно сократить время и затраты на его изготовление.

Языки управления фотоплоттерами

Почти все из существующих типов фотоплоттеров управляются посредством специального языка описания векторов. Обычно этот язык называют Gerber, который был изначально разработан компанией Gerber Scientific Company, а со временем стал промышленным стандартом (также известным, как RS-274). Не смотря на то, что этот язык эволюционирует вслед за развитием печатающего оборудования и технологий проектирования, разработчику в процессе планирования работы по генерации файлов в формате Gerber необходимо учитывать связанные с этим потенциальные трудности и ограничения.

Файл в формате Gerber описывает чертеж как последовательность команд черчения и координат в ASCII кодах. Команды черчения управляют используемой апертурой, включают или выключают источник света и т. д. Координаты определяют положение различных точек и линий на поверхности фотошаблона.

Структура Gerber файлов может меняться в зависимости от количества модификаций, которые были добавлены в формат в процессе развития вслед за изменением

возможностей печатающего оборудования. Организации, которая будет выполнять изготовление фотошаблона, могут понадобиться некоторые сведения, относящиеся к используемому системой Protel 99 SE формату Gerber, поэтому ниже приведено его краткое описание.

Gerber файлы пакета Protel состоят из записей кодов, разделенных символами возврата каретки и перевода строки. Каждая запись завершается символом "*".

Эти записи содержат абсолютные координаты и команды черчения, управляющие аперттурами. Таким образом, запись может состоять из следующего набора символов: **X800Y775***, что означает перемещение в точку с указанными координатами; или **D16**, что является командой на выбор новой аперттуры.

Некоторые фотоплоттеры резервируют коды **D01 — D09** не для выбора апертур, например:

- D01** Включение источника света.
- D02** Выключение источника света.
- D03** Вспышка источника света.

В некоторых старых плоттерах перед каждым изменением кода аперттуры должен быть послан специальный код **G54**. Последовательность команд в Gerber файле завершается специальной командой **M02***, за которой следует блок, содержащий символ hex 08, затем 509 пробелов (hex 20), а затем символы возврата каретки и перевода строки.

Любой Gerber файл может быть просмотрен с помощью текстового редактора, способного загружать "неформатированные" текстовые файлы.

Об аперттурах

Все фотоплоттеры, работающие в формате Gerber, используют аперттуры, которые описывают доступные инструментальные средства для прорисовки изображения на пленке. В случае векторных плоттеров эти аперттуры соответствуют различным размерам и формам отверстий в апертурном барабане, сквозь которые свет проецируется на светочувствительный слой пленки.

Растровые плоттеры не имеют ограничений на размер и форму апертур. Растровые системы формирования изображений интерпретируют данные об аперттурах, содержащиеся в Gerber файле, далее по ним синтезируется полный печатный образ, представляемый в виде растровой матрицы, который затем печатается построчно, подобно изображению на экране телевизора.

Использование апертур

Апертуры, используемые для трансляции файла печатной платы в набор Gerber файлов, сохраняются в файле с расширением **.APT**. Апертуры можно рассматривать как перья плоттера разной толщины. Описание аперттуры включает форму (например, квадрат со сторонами 50 мм) и действие (вспышка, штрих или любое другое).

Перед генерацией Gerber файла можно либо загрузить файл апертур, который соответствует возможностям используемого плоттера, либо позволить редактору печатных плат автоматически создать файл апертур на основе данных о примитивах (проводниках, контактных площадках и т. д.), расположенных в текущем файле печатной

платы. При работе с векторным фотоплоттером апертур, содержащиеся в **.APT** файле должны соответствовать апертурам, расположенным на его барабане. Растровые плоттеры используют файл апертур для перевода чертежных кодов непосредственно в карту изображения. При применении растровых устройств можно генерировать апертур из печатной платы и передавать таблицу этих апертур вместе с Gerber файлами. Организации, выполняющие работы по изготовлению фотошаблонов, как правило, предоставляют разработчикам апертурные таблицы своих векторных фотоплоттеров, а также подробные инструкции для генерации таблиц для растровых фотоплоттеров.

При использовании заранее заданного файла апертур редактор печатных плат проверяет все примитивы (проводники, контактные площадки и т. д.), содержащиеся в файле печатной платы и подбирает для них описания апертур из загруженного **.APT** файла. Если какому-либо примитиву не удастся подобрать идеально подходящей апертур, система прорисует его с помощью меньшей по размеру апертур. Если и таких апертур подобрать не удалось, генерируется файл с расширением **.MAT**, содержащий список недостающих апертур, а генерация Gerber файла прерывается.

При использовании векторного плоттера следует применять примитивы (размеры и формы проводников и контактных площадок) для которых существуют подходящие апертур. В противном случае это может привести к увеличению времени на изготовление фотошаблона за счет повторных проходов луча меньшей апертур для закрашивания нестандартных линий и полигонов.

Загрузка и редактирование апертур

Для загрузки, создания или редактирования используемых фотоплоттерами апертур необходимо выбрать команду меню **Design » Aperture Library**. После выполнения этой команды появляется диалоговое окно **Apertures** (рис. 6.83), где приведен список всех загруженных на данный момент апертур. При выводе чертежа все они будут использованы. Опции, расположенные в данном диалоговом окне, позволяют работать с новыми или уже существующими файлами апертур. Все сделанные изменения воздействуют на файлы апертур, которые в данное время загружены в память компьютера, чтобы сохранить их необходимо нажать кнопку **Save to APT File**.

- ◆ Пользователь имеет возможность определить до 1000 различных команд черчения в диапазоне **D00 — D9999**. Некоторые из этих кодов (обычно **D00 — D09**) могут быть зарезервированы плоттерами, поэтому их использовать не рекомендуется.
- ◆ При создании новых апертур в диалоговом окне **Draft Code Number** символ **D** вводить не нужно.

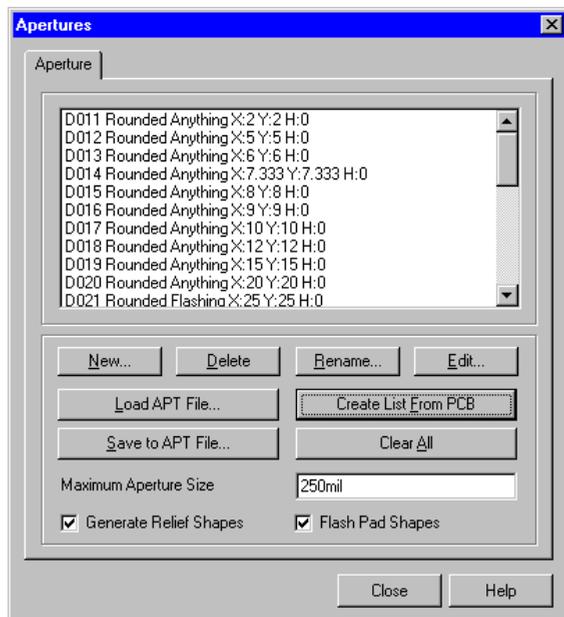


Рис. 6.83. Настройка апертур производится в диалоговом окне Apertures.

Печать в формате Postscript

На сегодняшний момент в полиграфии широко используются фотонаборные устройства высокого разрешения, получающие входные данные в формате Postscript. Это оборудование способно производить пленочные позитивы с разрешением до 2540 точек на дюйм и может стать достаточно дешевой альтернативой выводу в формате Gerber.

Однако, пользователи должны хорошо представлять себе ограничения этого способа печати чертежей печатных плат. Высокое разрешение этих систем вовсе необязательно означает высокую точность позиционирования или линейность, особенно при печати больших чертежей. Также существует ограничение на размеры используемых пленок. Выходные файлы в формате Postscript могут быть получены с помощью модуля печати чертежей на стандартных устройствах системы Windows, о чем подробно описано в разделе *Печать чертежа печатной платы средствами Windows*.

Советы по Postscript печати

Большинство Postscript принтеров и фотонаборных машин позволяет выполнять печать документов с разрешением от 300 до 2540 dpi. Вследствие высокого разрешения, получаемого с помощью этих устройств, многие пользователи ориентируются на использование Postscript печатающих устройств для выполнения чертежей, как деше-

вой альтернативы Gerber печати. Однако, здесь существуют некоторые ограничения, которые необходимо учитывать при выборе технологии печати.

Лазерные фотонаборные машины с высоким разрешением выводят изображение непосредственно на пленку или светочувствительную бумагу. Эти устройства выдерживают достаточную точность по горизонтали, но из-за неточностей работы механизма подачи пленки точность по вертикали может быть низкой.

В настоящее время некоторые фотонаборные машины используют не рулонную пленку, а резанную, закрепленную на специальном барабане. Линейная точность таких устройств выше, поэтому они являются неплохой альтернативой печати в формате Gerber для большинства проектов.

Для проверки линейной точности Postscript устройства необходимо создать файл с набором вертикальных и горизонтальных проводников известной длины и провести их тщательные измерения на выходном чертеже. Это позволит подобрать корректирующие масштабные коэффициенты для каждой из осей. Однако, линейная погрешность в разных местах большого чертежа может быть неодинаковой, поэтому необходимо тщательно проверять все конечные чертежи непосредственно перед передачей его в производство.

При печати чертежей на настольных лазерных принтерах с разрешением 300 или 600 dpi возникает проблема размазывания тонера, что может привести к неправильному изготовлению фотошаблона и появлению замыканий на плате. В этом случае рекомендуется на чертежах использовать большие зазоры между проводниками, а сам чертеж выводить под возможно большим увеличением.

Качество печати, получаемое при применении лазерных принтеров, во многом определяется используемой бумагой. На сегодняшний момент доступно несколько типов специальной бумаги (предназначенной главным образом для графической печати), которая уменьшает эффект растекания, что делает контуры более отчетливыми. Как правило, такая бумага несколько плотнее и имеет специальное покрытие, поэтому обращаться с ней намного проще. Единственное, что остается пользователю — это внимательно следить за чистотой такой бумаги.

Лазерные фотонаборные машины способны обеспечить печать чертежей с очень большим разрешением (до 2540 dpi) при больших размерах чертежа, например, А3 (или "В").

Некоторые Postscript принтеры отбрасывают текущие данные в случае не получения маркера конца страницы в установленный период времени. Это может привести к пропуску страниц при печати. Если такая проблема существует с Postscript принтером или любым другим печатающим устройством, необходимо изменить конфигурацию устройства таким образом, чтобы был параметр **Transmission Retry** (Повтор передачи) был равен 500 секунд (или любому другому большому значению). Как правило, этого времени достаточно для согласования скоростей печати и передачи данных, посылаемых программой **Print Manager**.

Печать с использованием перьевых плоттеров

Печать на перьевых плоттерах выполняется с использованием драйверов системы Windows. Как правило, они существуют для всех новых устройства, использующие

процедуры растровой, а не векторной печати, к которым относятся новые широкоформатные струйные плоттеры. На сегодняшний момент также доступны драйверы для векторных плоттеров, информацию о которых можно получить у непосредственно производителей плоттеров или у представителей компании Protel.

Передача изменений в проекте на этап разработки принципиальной схемы

Иногда нумерацию позиционных обозначений компонентов полезно производить на печатных платах, а не на схемах. Такой стиль упрощает поиск нужного компонента на плате. Редактор печатных плат может автоматически выполнить эту операцию (**re-annotation**), после чего все сделанные изменения можно передать обратно на этап разработки принципиальной схемы.

Повторное присваивание обозначений компонентов

В редакторе чертежей печатных плат обозначения компонентов могут быть пронумерованы на основе их местоположения (аналогично элементам на схеме). Для запуска процедуры обратного аннотирования используется команда меню **Tools » Re-Annotate**. На экране появляется диалоговое окно **Positional Re-Annotate**, в котором необходимо определить начальную точку и направление аннотирования. Существует четыре варианта направлений:

- Нумеровать снизу вверх, начиная с нижнего левого угла.
- Нумеровать сверху вниз, начиная с верхнего левого угла.
- Нумеровать слева направо, начиная с нижнего левого угла.
- Нумеровать слева направо, начиная с верхнего левого угла.

◆ При повторном обозначении компонентов их нумерация всегда непрерывна.

Процедура аннотирования использует "полосу" шириной 100 тысячных дюйма, двигаясь в направлении, указанном в каждой опции. Например, первая опция названа **Ascending X then Ascending Y** (сначала по оси X, затем по оси Y). Эта опция устанавливает полосу 100 тысячных дюйма по оси X, затем просматривает эту полосу по оси Y. Затем устанавливается следующая полоса по оси X и снова она просматривается по оси Y и т.д. Точкой привязки компонента считается левый нижний угол его контура. Сначала обрабатываются компоненты на верхнем слое, а затем на нижнем.

При выборе команды меню **Tools » Re-Annotate** изменения обозначений записываются в "было/стало" файл. Это простой ASCII файл, содержащий старые и соответствующие им новые обозначения. Для системы Protel 99 SE этот файл не является необходимым для передачи изменений в обозначениях обратно на этап проектирования принципиальной схемы, он просто обеспечивает запись сделанных изменений.

Обновление принципиальной схемы, исходя из изменений в печатной плате

После выполнения повторного присвоения позиционных обозначений на печатной плате информация об изменении обозначений может быть передана обратно на этап проектирования принципиальной схемы. Для этого достаточно выполнить одну команду меню **Design » Update Schematic**. Синхронизатор проекта откроет все листы схемы и обновит их согласно сделанным изменениям обозначений компонентов. Отметим, что нет никаких ограничений на количество циклов переобозначений, выполненных до обновления схемы. Если необходимо, можно выполнить изменение обозначений на печатной плате в ручном режиме и затем описанным способом, то есть с помощью команды **Design » Update Schematic**, обновить принципиальную схему.

Кроме передачи изменений в обозначениях, синхронизатор пакета системы Protel 99 SE также может пересылать другую информацию об изменениях, сделанных на печатной плате, на этап проектирования схемы. Поддерживаются следующие изменения:

- Изменения обозначений компонентов
- Изменения комментариев к компонентам
- Изменения посадочных мест
- Удаления компонентов

Синхронизатор сравнивает информацию о связях на схеме и на плате. Он не изменяет соединения в схеме, но любые изменения связанности цепей, сделанные на плате, заносятся в специальный отчет. Для создания такого отчета необходимо нажать кнопку **Report**, расположенную в нижней части вкладки **Changes**. Затем этот отчет можно использовать в качестве контрольного списка для обновления связей в схеме.

При выборе команды меню **Design » Update Schematic** на экране появляется диалоговое окно **Update**, в заголовке которого указано название проекта принципиальной схемы. Для каждого изменения, которое может быть передано в схему, создается макрос (рис. 6.84). После нажатия кнопки **Execute** все компоненты, расположенные на листах принципиальной схемы, у которых были изменены атрибуты, автоматически обновляются.

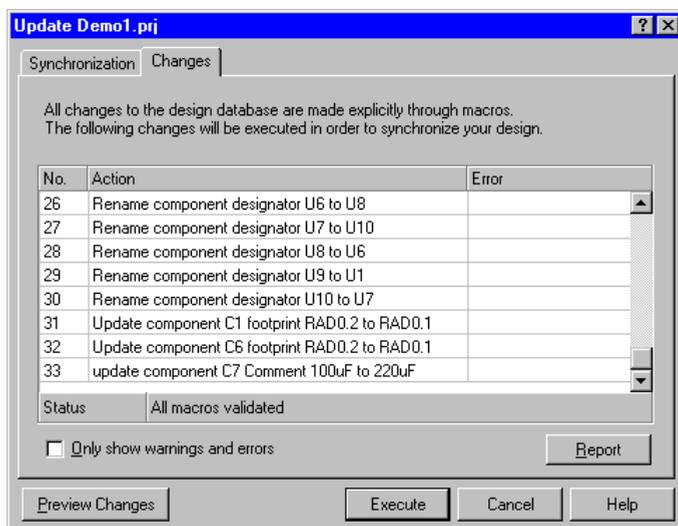


Рис. 6.84. Синхронизатор проекта передает изменения, сделанные в редакторе печатных плат обратно в редактор принципиальных схем.

Интерфейс с другими системами проектирования

Редактор печатных плат поддерживает несколько способов обмена проектной информацией с другими системами проектирования.

Если система Protel 99 SE используется для проектирования как принципиальной схемы, так и печатной платы, тогда отпадает необходимость создания списка соединений для передачи проектной информации. Этот пакет включает мощный инструмент синхронизации проекта, который автоматизирует процедуру передачи проектной информации из схемы на плату и обратно. Подробное описание работы синхронизатора приведено в разделе *Передача информации о схеме в редактор печатных плат* главы *Разработка принципиальных схем*, а также в разделе *Передача изменений в проекте на этап разработки принципиальной схемы* главы *Проектирование печатных плат*.

Загрузка списка соединений

Файлы списков соединений могут быть записаны в нескольких различных форматах, но обычно используется текстовый ASCII формат, в котором содержится, по крайней мере, два вида информации:

1. Описания компонентов в проекте.
2. Список всех межвыводных соединений в проекте.

Некоторые форматы объединяют оба типа данных в одном описании, другие, включая и собственный формат системы Protel, разделяют разные типы данных на отдельные секции.

Так как ASCII файлы списков соединений имеют обычный текстовый формат, они могут быть без труда преобразованы в файлы других форматов с помощью простых программ-конвертеров. Список соединений также может быть создан (или скорректирован) вручную с использованием любого текстового редактора.

Редактор печатных плат умеет загружать списки соединений в форматах Protel, Protel2 и Tango.

Загрузка списка соединений

Для загрузки списка соединений в редактор печатных плат используется команда меню **Design** » **Netlist**. Появится диалоговое окно **Load/Forward Annotate Netlist**, в котором нужно нажать кнопку **Browse**, выбрать нужный список соединений и нажать кнопку **OK** (рис. 6.85).

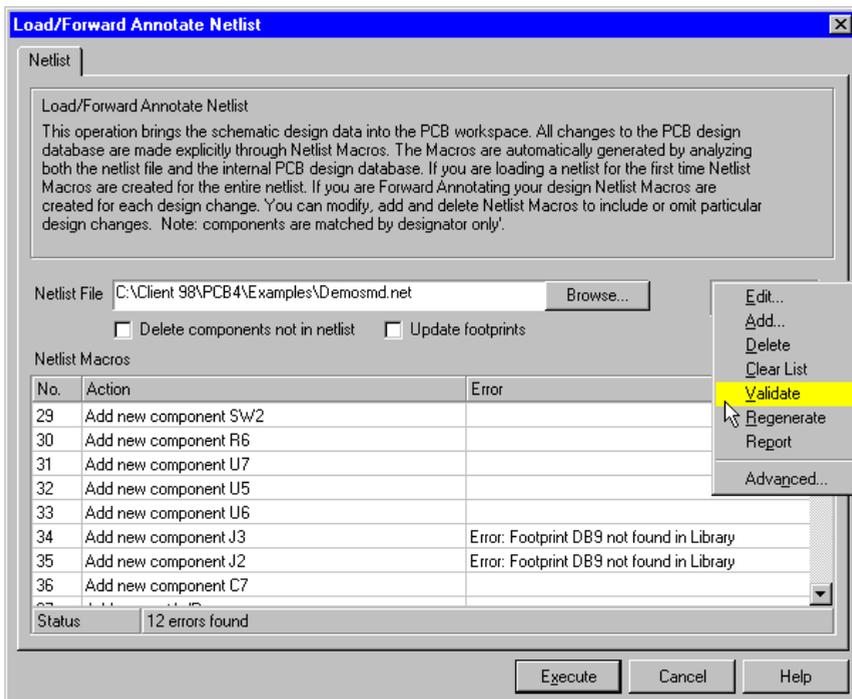


Рис. 6.85. Загрузка списка соединений.

После этого редактор проанализирует выбранный список соединений и существующий проект печатной платы. Для всех обнаруженных несоответствий будут созданы макросы **Netlist Macro**, с помощью которых редактор печатных плат сможет выполнить действия по обновлению текущего чертежа для приведения его в соответствие с выбранным списком соединений.

Если загрузка списка соединений выполняется первый раз, макросы создаются для всего списка. Если же производится прямое аннотирование, макросы создаются только для каждого изменения в проекте.

Перед запуском на выполнение, осуществляемым с помощью кнопки **Execute**, рекомендуется проверить все макросы на предмет соответствия их требуемым действиям.

Работа с макросами синхронизатора проекта

После того, как редактор печатных плат проведет анализ списка соединений и данных в проекте платы, для каждого обнаруженного несоответствия будет создан макрос. Список созданных макросов отображается в диалоговом окне **Load/Forward Annotate Netlist**, в порядке их последующего выполнения. В число возможных операций, выполняемых макросами, входят:

- Удаление узлов.
- Удаление цепей.
- Удаление компонентов.
- Добавление компонентов.
- Добавление цепей.
- Изменение имен цепей.
- Изменение посадочных мест компонентов.
- Изменение обозначений компонентов.
- Добавление узлов.
- Изменение комментариев к компонентам.

Кроме возможности проверять, что именно произойдет после загрузки списка соединений, пользователь также может добавлять, редактировать и удалять макросы синхронизатора проекта.

Добавление, редактирование и удаление макросов (Edit, Add и Delete)

С помощью этих операций можно исключить или выполнить определенные изменения в проекте. Они также используются, когда нужно изменить список соединений платы, не возвращаясь на этап проектирования схемы. Ниже перечислены пункты, которые нужно учитывать при редактировании макросов:

- Следует придерживаться одного регистра при наборе текста для обозначений и цепей.
- Узлы в списке цепей определяются парой "обозначение компонента – номер вывода" (например, **J3-2**). Согласно требованиям редактора библиотек топологических посадочных мест номер вывода может иметь максимум четыре символа (буквы или цифры) без пробелов.

- Посадочные места должны соответствовать именам в библиотеках, подключенным в данный момент к системе.
- После выполнения макросов рекомендуется обновить содержимое экрана.

Проверка правильности макросов (Validate)

После редактирования макросов необходимо проверить их правильность. Это особенно важно, если макросы созданы вручную. Выполнение команды Validate приведет к проверке макросов редактором печатных плат на предмет возможности их выполнения и выдаче отчета о найденных ошибках.

Повторная генерация макросов (Regenerate)

При выполнении команды повторной генерации редактор удаляет все существующие макросы, проводит повторный анализ списка соединений и существующего проекта платы, после чего создает новые макросы.

Исправление ошибок в макросах синхронизатора проекта

Перед запуском макросов на выполнение рекомендуется исправить все найденные в них ошибки. Ниже приводятся описания возможных ошибок с указанием макросов, в которых они могут произойти, и возможных их причин.

Net not found (цепь не найдена)

Макрос делает попытку: добавить или удалить узел; удалить цепь или изменить имя цепи, когда данная цепь отсутствует в существующем проекте платы.

Component not found (компонент не найден)

Макрос делает попытку: добавить или удалить узел, когда обозначение компонента в макросе указано неверно или компонент отсутствует в существующем проекте платы; удалить компонент или изменить его топологическое посадочное место, обозначение или комментарий, когда он отсутствует в проекте платы.

Node not found (узел не найден)

Макрос делает попытку: добавить или удалить узел из компонента, который не имеет такого вывода; или удалить узел, который не существует в указанной цепи.

Net already exists (цепь уже существует)

Макрос делает попытку: добавить в проект цепь, когда цепь с таким именем в существующем проекте платы уже существует.

Component already exists (компонент уже существует)

Макрос делает попытку: добавить в проект компонент, когда компонент с таким обозначением в существующем проекте платы уже существует.

New footprint not matching old footprint (новое посадочное место не соответствует старому)

Макрос пытается изменить посадочное место компонента, когда задействованные на старом посадочном месте выводы не соответствуют аналогичным выводам на новом. Такое может произойти, если новый компонент имеет меньше выводов или их нуме-

рация в списке соединений (которая берется из схемы) отличается от нумерации выводов компонента на плате.

Footprint not found in Library (посадочное место в библиотеке не найдено)

Макрос пытается добавить новый компонент или изменить посадочное место существующего компонента, когда оно отсутствует во всех подключенных библиотеках, и не найдено альтернативных библиотечных ссылок в файле перекрестных ссылок (**ADVPCB.XRF**).

Alternative footprint used instead (предупреждение: будет использовано альтернативное посадочное место)

Макрос делает попытку добавить новый компонент или изменить посадочное место компонента, когда оно отсутствует во всех подключенных к системе библиотеках. Но в файле перекрестных ссылок (**ADVPCB.XRF**) обнаружена альтернативная библиотечная ссылка и найденный компонент будет загружен из одной из библиотек, расположенных в списке текущих библиотек. Перед тем, как выполнить макрос с таким предупреждением, рекомендуется проверить, что найденное альтернативное посадочное место соответствует требованиям пользователя.

Когда макрос пытается загрузить или изменить посадочное место компонента, которое отсутствует в подключенных библиотеках, он использует поле комментариев этого компонента для поиска файла перекрестных ссылок (**ADVPCB.XRF**). В этом файле перечислены отсортированные по типу компоненты и соответствующие им посадочные места. Например, если компонент **U1** — это микросхема **74LS00**, но пользователь забыл включить для нее посадочное место, тогда при проверке макроса, добавляющего данный компонент, будет выполнен поиск корпуса микросхемы в **XRF** файле. **74LS00** имеет тип корпуса **DIP14**, который и будет загружен из какой-либо подключенной библиотеки.

Резюме

Большинство проблем с загрузкой списка соединений схемы можно разделить на две категории.

1. В списке соединений отсутствует информация о топологическом посадочном месте компонента; пользователь забыл подключить требующуюся библиотеку посадочных мест к системе (**Design » Add/Remove Library**); указанное в списке соединений посадочное место не доступно ни в одной подключенной библиотеке.
2. Новое топологическое посадочное место не соответствует старому, причиной этого обычно является различная нумерация выводов компонента на схеме и на печатной плате.

Библиотеки элементов схем содержат изображения специфических компонентов и устройств. Библиотеки топологических посадочных мест содержат общие посадочные места, которые могут принадлежать компонентам различных типов, имеющих разное распределение выводов.

Например, посадочное место транзистора может представлять различную комбинацию выводов эмиттера (**Е**), базы (**В**) и коллектора (**К**), каждому из которых в редакторе печатных плат должен быть назначен правильный номер. С диодами похожая си-

туация — они имеют выводы, которым в схеме присваиваются имена **A** (анод) и **K** (катод).

В этом случае можно либо поменять номера выводов посадочного места на плате, либо изменить их у компонента на схеме так, чтобы они соответствовали друг другу.

Поддерживаемые форматы списка соединений

Редактор печатных плат пакета Protel 99 SE может загружать списки соединений в формате Protel и Protel2. Подробно синтаксис этих форматов приведен в разделе *Обмен данными с другими системами проектирования* главы *Разработка принципиальных схем*.

Списки соединений, созданные с помощью других программ разработки принципиальных схем имеют очень похожий формат. Однако, порядок, в котором выводится информация о компоненте или цепи, может отличаться от принятого в системе Protel, и имя топологического посадочного места, позиционное обозначение компонентов, содержимое полей и идентификаторы выводов может потребовать ручного редактирования.

Ограничения на параметры списка соединений

Содержимое полей **Designators** и **Package Descriptions** не должно превышать 12 символов. Поле **Part Type** может содержать до 32 символов. Имена цепей **Net Names** могут иметь максимум 20 символов. Номера выводов в списке соединений должны содержать не более 4 символов. Использование пробелов в этих полях не допускается.

Ограничений на число компонентов или узлов списка соединений не существует.

Экспорт списков соединений

Существует два метода экспорта данных о текущем проекте в виде списка соединений: экспорт внутреннего списка или его генерация на основе имеющегося чертежа платы. Оба метода реализованы в модуле **Netlist Manager**, запускаемом командой меню **Design » Netlist Manager**.

Экспорт внутреннего списка соединений

В этом случае экспортируется внутренний список соединений проекта, содержащий информацию о связанности, полученную из редактора принципиальных схем. Если трассировка проекта не производилась, связанность представляется виртуальными линиями связи. Для экспорта списка соединений необходимо нажать кнопку **Menu**, расположенную в диалоговом окне **Netlist Manager**, из появившегося всплывающего меню выбрать команду **Export Netlist from PCB**. Список соединений создается внутри базы данных проекта.

Генерация списка соединений на основе имеющейся трассировки

При использовании этого метода генерируется список компонентов и связей (то есть список соединений) из набора компонентов, проводников и переходных отверстий на плате. Для генерации списка соединений необходимо нажать кнопку **Menu**, распо-

женную в диалоговом окне **Netlist Manager**, из появившегося всплывающего меню выбрать команду **Create Netlist from Connected Copper**. Список соединений создается внутри базы данных проекта.

Модули импорта и экспорта файлов

Редактор чертежей печатных плат имеет возможность экспорта проекта в форматах старых версий системы Protel и формате DWG/DXF системы AutoCAD, а также импорта файлов старых версий системы Protel, файлов в формате Gerber и проектов печатных плат системы OrCAD Layout.

Для импорта необходимо выполнить команду меню **File » Import** и в появившемся окне **Import File dialog** выбрать нужный тип файлов.

Работа с файлами старых версий системы Protel

В системе Protel 99 SE применен новый формат файлов печатных плат PCB 4.0. Для сохранения проектов печатных плат в старом формате PCB 3.0, который использовался в системах Advanced PCB V3, Protel 98, Protel 99 и Protel99 SP1, необходимо выполнить команду меню **File » Save Copy As**.

Импорт проектов старых форматов в базу данных проекта системы Protel 99 SE осуществляется обычным способом, например, перетаскиванием PCB файла из Проводника системы Windows в среду Design Explorer и последующим щелчком на нем левой кнопкой мыши.

Интерфейс с механическими САПР

Система Protel 99 SE имеет двунаправленный интерфейс связи с механическими САПР через файлы форматов DWG и DXF. Для запуска процессов импорта или экспорта используются команды меню редактора печатных плат **File » Import** или **File » Export** соответственно.

Функции интерфейса связи с механическими САПР включают:

- Полная поддержка импорта и экспорта файлов в формате DWG и DXF версий начиная с 2.5 и заканчивая R14.
- При импорте автоматически запускается диалоговое окно **Import from AutoCAD**, в котором устанавливается соответствие импортируемых слоев существующим слоям платы.
- Импорт из системы AutoCAD размера чертежа.
- Автоматическое масштабирование при импорте.
- Преобразование компонентов в блоки и наоборот.
- Поддержка метрической и дюймовой систем единиц.

Более подробная информация по вопросу импорта и экспорта файлов этих форматов приведена в интерактивной справочной системе программы.

Интерфейс с системой OrCAD Layout

Система Protel 99 SE имеет возможность импорта проектов в бинарном формате, созданных системой проектирования OrCAD Layout V9.x. Для импорта таких файлов необходимо выполнить команду меню **File » Import** и выбрать тип файла с расширением **.MAX**.

Функции импорта включают:

- Непосредственная загрузка проекта платы OrCAD Layout (V9.x) в редактор чертежей печатных плат.
- При импорте автоматически запускается диалоговое окно **OrCAD Layout Importer**, в котором устанавливается соответствие импортируемых слоев существующим слоям платы.
- Создание отчета о выполненной операции импорта.
- Импорт топологических библиотек с расширением **.LLB** в редактор библиотек топологических посадочных мест системы Protel 99 SE.
- Импорт библиотек непосредственно из чертежа печатной платы.

Более подробная информация по вопросу импорта и экспорта файлов этих форматов приведена в интерактивной справочной системе программы.

Глава 7

Программа CAMtastic! 2000 Designers' Edition

Общие сведения

Программа CAMtastic! 2000 Designers' Edition входит в комплект поставки sbcncvs Protel 99 SE и представляет собой специализированный инструмент подготовки управляющих файлов для производства печатных плат. Программа загружает и автоматически распознает форматы Gerber (RS274-D, RS274X или Fire9000), файлы AutoCAD DXF/DWG, известные форматы HPGL/HPGL2, Excellon Drill, Mill или Route-файлы. Модуль импорта DXF/DWG поддерживает все объекты современных версий AutoCAD, включая LWPolylines, 3D-Faces, M-Lines, Polyline Spline, Hatches и Mtext.

Для описания апертур в программе CAMtastic доступны около тысячи D-кодов, причем список апертур проекта создается автоматически при импорте файла. Различные стандартные и пользовательские наборы апертур могут сформированы и сохранены в виде специальных файлов с таблицами апертур, что делает возможным их повторное использование, упрощая, тем самым, обработку проектов. Программа поддерживает закрашенные полигоны и пустоты в них, истинные дуги, композитные объекты и перекодирование.

Вся информация о проекте (слои, результаты выполнения запросов, сообщения о проверке правил проектирования DRC и информация из списка соединений) отображается на панели управления. Эта панель обеспечивает быстрый доступ к таблицам, настройкам программы и функциям генерации отчетов, например, сводного отчета для размещения проекта платы на изготовление. Местоположение панели управления может быть без труда изменено в соответствии с желанием пользователя.

Программа CAMtastic! 2000 Designers' Edition имеет удобный набор инструментов для масштабирования и панорамирования окна просмотра проекта. Здесь имеются средства генерации списка соединений и наборов правил проектирования (DRC) из Gerber данных. Для отдельных объектов или их групп может быть получена и отображена самая разнообразная информация, например: используемый D-код, форма, размеры, тип, протяженность и принадлежность цепи. Высокая точность измерения расстояния между двумя точками достигается использованием десяти различных настроек сеток. Имеется возможность выявления различий между содержимым двух слоев, а также возможность просмотра прозрачных слоев.

Любая апертура может быть использована для прорисовки так называемых флэшей — отдельных объектов, повторяющих форму апертуры и получаемых разовым включением лампы (вспышкой) для рисования линий, многоугольников, текста (включая

шрифты TrueType) или дуг. Имеется возможность настройки отображения дуг на экране набором линий, что позволяет значительно ускорить обновления экрана, без потери их истинного описания в файле. В зависимости от собственных нужд и предпочтений пользователь может использовать различные методы рисования. Изменение шага сетки позволяет обеспечить требуемую точность прорисовки объектов.

В программе имеются разнообразные средства редактирования и выделения графических объектов. У флэшей могут быть изменены размеры и форма. У линий могут быть изменены местоположение и ширина. Несколько линий могут быть объединены в полилинии, и наоборот, полилинии можно разбить на несколько отдельных сегментов, а к ним добавлены новые вершины и скосы. Имеется возможность удаления, копирования, вставки, зеркального отображения, масштабирования и изменения апертуры выделенных объектов. Слои могут быть переименованы, скопированы, выровнены, изменены в размере или удалены.

Значительно повысить эффективность работы пользователя позволяют имеющийся в программе набор дополнительных инструментальных средств — мастеров, с помощью которых выполняется загрузка внешних таблиц апертур, импорт файлов в формате DXF/DWG, панелизация фотошаблонов на листе пленки заданного размера, преобразование групп идентичных объектов в один флэш. Имеется возможность заливки замкнутых контуров, получения каплевидной формы контактных площадок (teardrop), генерации внешнего контура платы. Композитные слои позволяют прорисовывать объекты нестандартной формы.

Имеются средства улучшения технологичности проекта. В программе предусмотрена возможность панелизации загруженных и отредактированных топологий печатных плат с оптимизацией выходного управляющего кода за счет использования операторов циклов, что позволяет избежать неконтролируемого увеличения размера выходного файла. Для создания контуров фрезерования можно определить до десяти различных стратегий. В специальной таблице инструментов могут быть описаны до 99 различных инструментов, при этом будут учитываться поля компенсации диаметра инструмента, врезная подача, скорость вращения шпинделя, глубина реза.

Специальный набор инструментов позволяет анализировать топологию платы и оценивать стоимость ее изготовления. В подробных отчетах может быть отражена следующая информация: критические величины (минимальная ширина линии и зазор между проводниками), общие характеристики платы (габаритные размеры, площадь, периметр, число вырезов), контактные площадки (минимальный размер и шаг, общее количество), информация о сверлении отверстий с учетом пар слоев для изготовления глухих и скрытых переходных отверстий. Имеется возможность расчета площади золочения и лужения.

Интерфейс программы CAMtastic! 2000 Designers' Edition ничем не отличается от большинства других Windows-программ. Любое действие здесь выполняется несколькими способами. Например, команда меню может быть выбрана с помощью мыши или горячей клавишей (подчеркнутой в меню буквы) при нажатой клавише **ALT**. В диалоговых окнах также возможна работа только с клавиатурой: переход с одной опции на другую осуществляется нажатием клавиши **TAB** или **SHIFT+TAB**, нажатие клавиш **SPACEBAR** или **ENTER** активизирует нужную кнопку или поле. Кнопки на панелях инструментов предлагают быстрый доступ к различным функциям программы, большинству из которых также назначены горячие клавиши. Интерактивная

справочная система вызывается с помощью команд из раздела меню **Help**. Нужную информацию здесь можно найти по содержанию или средствами поиска ключевых слов. Полезные советы могут автоматически выводиться при каждом запуске программы или выборе команды **Help » Tip of the Day**. При работе в диалоговом окне можно получить справку по всем его элемента нажатием клавиши **F1**.

Инсталляция программы

Установка программы SAMtastic! 2000 Designers' Edition на персональный компьютер достаточно проста и не требует никакой специальной подготовки.

Системные требования соответствуют следующим требованиям и рекомендациям:

Рекомендуемая конфигурация:

- Операционная система Windows NT4/2000 Professional
- Персональный компьютер на базе процессора Pentium III
- 128 Мб оперативной памяти
- 200 Мб дискового пространства
- Монитор с разрешением 1024x768 пикселей
- 32 разрядная цветовая палитра
- Устройство считывания CD-ROM
- Мышь или аналогичное устройство управления

Минимальная конфигурация

- Операционная система Windows 95/98/2000/Me
- Персональный компьютер на базе процессора Pentium 166 MHz
- 32 Мб оперативной памяти
- 40 Мб дискового пространства
- Монитор с разрешением 800x600 пикселей
- Палитра из 256 цветов
- Устройство считывания CD-ROM
- Мышь или аналогичное устройство управления

Инсталляция программы SAMtastic! 2000 Designers' Edition ничем не отличается от установки большинства Windows-приложений. Для выполнения полной установки необходимо выбрать опцию Typical Installation.

ОСНОВНЫЕ ПОНЯТИЯ

Рассмотрим основные приемы работы в программе CAMtastic! 2000 Designers' Edition, являющихся общими для всех проектов. На рис 7.1 представлен общий вид окна программы с загруженной платой.

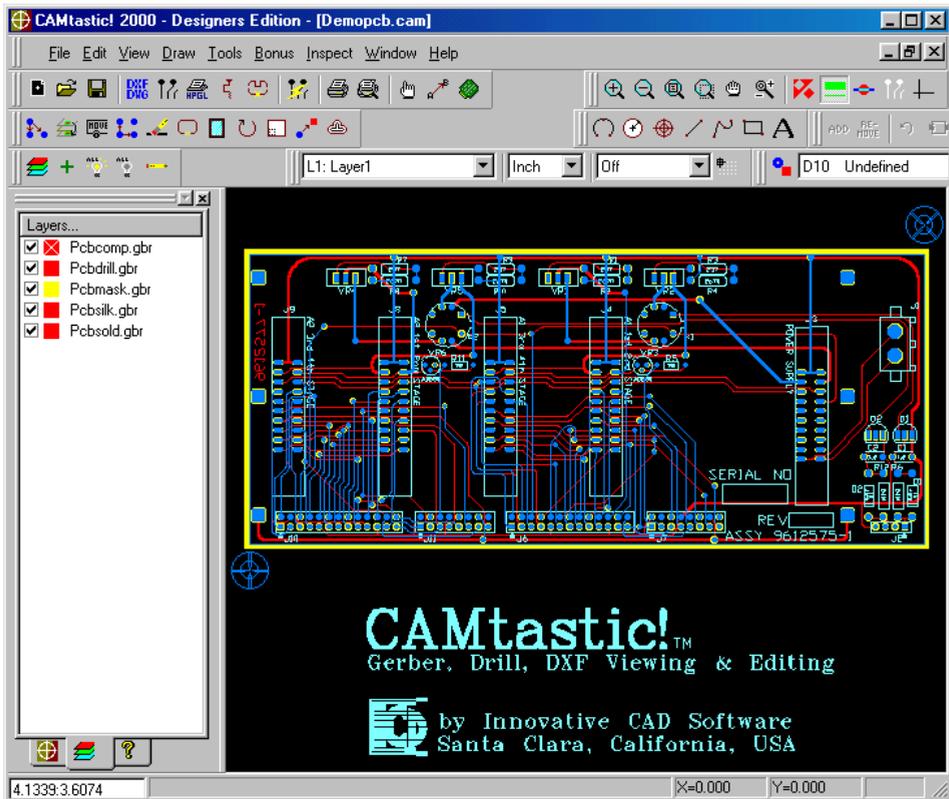


Рис. 7.1. Общий вид окна CAMtastic! 2000 Designers' Edition с загруженной платой.

Программа имеет одиннадцать панелей инструментов, каждая из которых (кроме меню команд) может быть выключена или перемещена в любое место экрана. Панели инструментов содержат более 70 различных кнопок и 4 выпадающих списка.

После запуска команды меню **View** » **Toolbars**, на экране появится диалоговое окно **Customize** (рис. 7.2), где на вкладке **Toolbars** будет представлен полный список имеющихся в системе панелей инструментов. Кроме того, здесь имеется возможность соз-

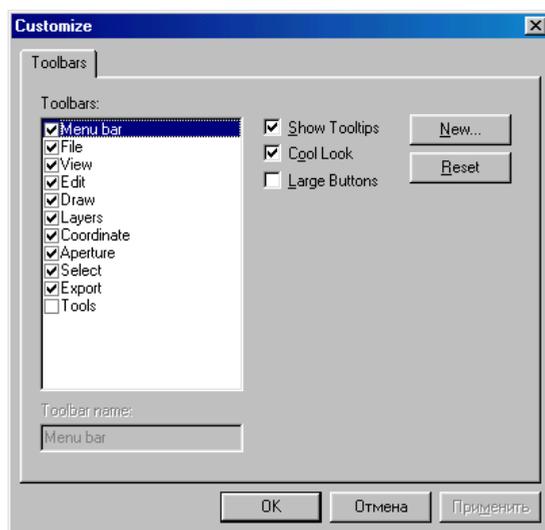


Рис. 7.2. Настройка панелей инструментов.

дания новых панелей, добавления и удаления кнопок, восстановления настройки по умолчанию в случае внесения нежелательных изменений.

Опция **Show ToolTips** управляет отображением всплывающих подсказок, появляющихся при наведении указателя мыши на нужную кнопку. Опция **Cool Look** управляет стилем отображения панелей инструментов.

Для создания новой панели инструментов, следует нажать кнопку **New** и ввести ее имя. После нажатия кнопки **OK**, это имя будет добавлено в список панелей инструментов, представленный в окне **Customize**. После закрытия этого диалогового окна на экране появится новая, пустая панель инструментов.

Добавление кнопок на новую панель осуществляется методом перетаскивания при открытом окне **Customize**. При перетаскивании кнопки в любое другое место экрана (не на панель инструментов) кнопка будет удалена. Если диалоговое окно **Customize** закрыто, то перетаскивание кнопки осуществляется при нажатой клавише **ALT**. Если на какой-либо из стандартных панелей были сделаны нежелательные изменения, исходные настройки можно восстановить, выделив эту панель инструментов и нажав кнопку **Reset**. Следует



Рис. 7.3. Панель Project Workspace закладка Main View.

обратить внимание, что при выборе пользовательской панели инструментов кнопка **Reset** меняется на кнопку **Delete**.

Панель управления Project Workspace

Панель управления **Project Workspace** представляет собой весьма удобный инструмент доступа к различной проектной информации и инструментальным средствам, необходимым во время текущей сессии работы с проектом. Размеры панели могут быть изменены, а сама она перемещена в любое место окна программы. На панели Project Workspace (рис. 7.3) могут присутствовать до пяти закладок: **Main**, **Layer**, **Info**, **DRC/DFM Check** и **Netlists**.

На закладке **Main View** отображаются таблицы данных, отчеты, сообщения и свойства объектов, а также основные настройки, единые для любого сеанса работы с программой. Здесь присутствуют также ссылки на вызов интерактивной справочной системы и вызов информации о текущей версии программы SAMtastic! 2000 Designers' Edition.

Любой элемент на панели управления может быть открыт двойным щелчком левой кнопки мыши.

На закладке **Layer View** (рис. 7.4) отображаются все слои активного проекта. При включении данного слоя, его содержимое будет отображаться в окне проекта. Щелчок на цветовом поле рядом с именем слоя вызывает окно настройки цвета.

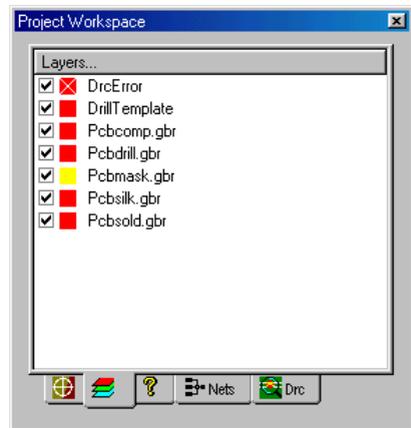


Рис. 7.4. Закладка Layer View.

Двойной щелчок левой кнопкой мыши на любом слое делает его активным (текущим), о чем сигнализирует крестик на цветовом соответствующем поле. В приведенном на рисунке примере все слои включены, но только один из них (последний) является активным.

Щелчок правой кнопкой мыши на имени любого слоя вызывает контекстное меню, предлагающие пользователю ряд дополнительных возможностей:

Type: Показывает, какой тип (если есть) назначен этому слою в таблице слоев программы SAMtastic! 2000 Designers' Edition.

ON: Признак и команда включения слоя на отображение.

Current: Признак и команда выбора данного слоя активным (текущим). Такого же результата можно достичь, сделав двойной щелчок левой кнопкой мыши на имени слоя на панели управления.

Lock: Признак и команда блокировки редактирования данного слоя.

Delete: Команда удаления данного слоя (это действие не может быть отменено).

Rename: Команда переименования данного слоя.

Open with Notepad: Команда на открытие файла, информация из которого была импортирована на данный слой. Если такого файла нет, например, объекты копировались из другого слоя, будет выдано сообщение об ошибке. Сообщение будет появляться до тех пор, пока не будет выполнена операция экспорта данного слоя во внешний файл (команда **File » Export**)

Reload File: Команда повторной загрузки на данный слой информации из внешнего файла (Drill/Rout/Mill или Gerber) для отказа от нежелательных изменений.

На закладке **Info View** (рис. 7.5) отображается информация, относящаяся к выполненным запросам и измерениям. Если отображаемая информация не помещается в окне, появляются линейки прокрутки. Вся информация об одном объекте группируется в папку.

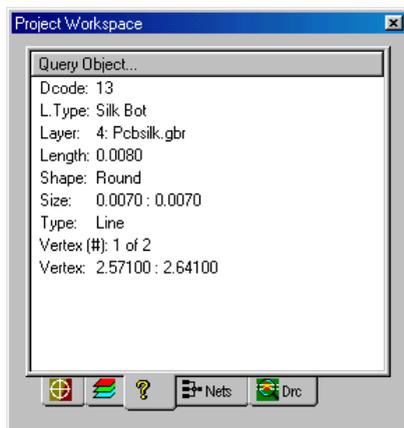


Рис. 7.5. Закладка Info View.

Двойной щелчок левой кнопкой мыши на значке папки открывает или закрывает данную папку.

На закладке **NetList View** (рис. 7.6) отображается список цепей, полученный с помощью команды экстракции списка соединений из соединенных участков меди **Netlist Extraction**. После щелчка левой кнопкой мыши на любой из цепей она будет подсвеченной в окне проекта. Масштаб отображения изменится таким образом, чтобы выбранная цепь отображалась оптимально. Для обновления этого списка используется команда **Inspect » View Nets**.

По умолчанию эта закладка на панели управления не отображается, но если в проекте хоть раз выполнялась команда экстракции списка соединений, то она автоматически появляется.

На закладке **DRC/DFM View** (рис. 7.7) отображается список нарушений, выявленных в ходе проверки общих правил проектирования (DRC) и технологических норм (DFR) для данного проекта.

Все нарушения сгруппированы в папки и имеют иерархическую структуру. Каждая папка верхнего уровня соответствует определенному правилу, внутри нее перечислены все нарушения данного

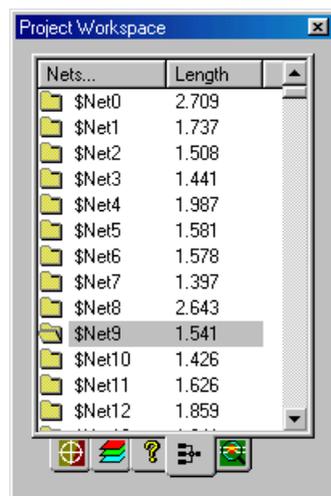


Рис. 7.6. Закладка NetList View.

правила. При выборе любой из папок нижележащего уровня, в окне проекта будут показаны объекты, вызвавшие данное нарушение. Для обновления списка используется команда **Inspect » View Errors**.

По умолчанию эта закладка на панели управления не отображается, но если в проекте хоть раз выполнялась проверка правил DRC/DFM, то она автоматически появляется.

Строка состояния

Отображение строки состояния программы CAMtastic! 2000 Designers' Edition включается командой **View » Status Bar**. Строка состояния всегда располагается в нижней части окна программы и содержит информацию об активном проекте.

В левом углу строки состояния в поле **Coordinate Box** отображаются координаты указателя мыши, в правом — максимальные размеры топологии в текущих единицах измерения и число выделенных объектов. В центральной части строки состояния приводится текстовое описание выполняемой операции. Если ни одна команда не была запущена, текстовая строка будет пустой.

Максимальные размеры топологии автоматически обновляются по мере работы с проектом, что дает возможность пользователю постоянно контролировать размер проекта.

Поле **Coordinate Box** выполняет в программе две функции: отображение координат указателя мыши и ввод координат указателя мыши при выполнении операций рисования и редактирования в формате X:Y (до 6 цифр).

Рассмотрим выполнение команды **Draw » Line**, предназначенной для прорисовки линий. Вместо того, чтобы выбирать начальную точку линии с помощью мыши, следует щелкнуть левой кнопкой мыши на поле координат и ввести значение 1:1. Для завершения операции необходимо нажать клавишу **Enter**. Удалим в поле координат старое значение, введем новое - 1:2 и нажмем клавишу **Enter**. В окне проекта появится вертикальная линия, нарисованная на активном слое от точки с координатами (1,1) до точки с координатами (1,2) в текущих единицах измерения. Таким способом рисуется последовательный набор линий. Для завершения операции следует нажать клавишу **Esc** или правую кнопку мыши. Такой метод ввода называется вводом абсолютных координат.

Окно **Coordinate Box** допускает также использование относительных координат, отсчитываемых от начальной точки, записанных в линейных или полярных координатах.

Для рисования прямой линии из начальной точки, выполним команду **Draw » Line** и зададим начальную точку посредством введения абсолютных координат или с помо-



Рис. 7.7. Закладка DRC/DFM View.

щью мыши. Затем введем в поле координат значение @1,0 и нажмем клавишу Enter. В окне проекта будет нарисована линия длиной в одну единицу измерения от начальной точки. Рисование линий с использованием полярных координат выполняется аналогично, только в поле следует ввести значение @1<45. В окне проекта будет нарисована линия длиной в одну единицу измерения под углом в 45° к горизонтальной оси.

Относительные полярные координаты записываются в формате @A<B (точка, расположенная от начальной точки на расстоянии A и под углом B относительно горизонтальной оси).

Абсолютные координаты записываются в формате X:Y (точка с координатами X и Y). Относительные линейные координаты записываются в формате @X:Y (точка, расположенная на расстоянии X и Y от начальной точки). Тип координат определяется форматом записи в окне **Coordinate Box**.

Работа с мышью

Рассмотрим некоторые особенности работы с мышью в системе CAMtastic. Как и в любой другой программе Windows, в системе левая кнопка мыши здесь используется для выбора, выделения и перемещения объектов. При перемещении мыши соответствующий выбранный инструмент или указатель мыши перемещается на экране.

Привычный указатель мыши в виде стрелки используется для стандартных операций в системе Windows, таких как выбор команд меню и кнопок на панелях инструментов и опций в диалоговых окнах.

Для удобства пользователей в программе CAMtastic используется система контекстных меню, состав команд которых зависит от выполняемой в данный момент операции. Если в данный момент ничего не рисуется и не редактируется, то щелчок правой кнопкой мыши в окне проекта вызовет стандартное контекстное меню, дающее доступ к различным функциям масштабирования. Здесь же содержится команда отмены или повтора выполнения предыдущей операции.

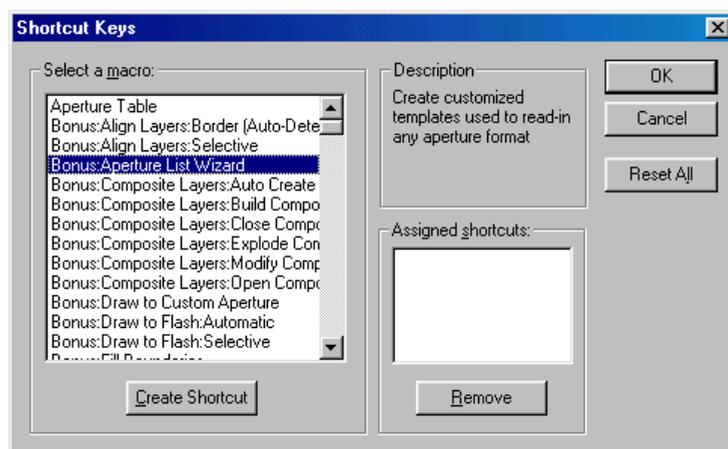


Рис. 7.8. Окно настроек горячих клавиш.

Многие действия в программе CAMtastic, такие как рисование или редактирование, требуют последовательного нажатия различных клавиш и кнопок мыши. Как правило, щелчок правой кнопкой мыши завершает выполнение текущей операции (в отличие от нажатия клавиши **Esc**, которая отменяет данное действие).

Горячие клавиши

Пользователям, которые используют для вызова различных команд не мышь, а клавиатуру, в программе предлагается система комбинаций горячих клавиш. Организована она аналогично большинству других программ Windows, где нажатие подчеркнутой в названии раздела меню буквы при удержании клавиши **ALT**, раскрывает раздел, а последующее нажатие другой клавиши запускает соответствующую команду.

В меню рядом с именем команды приводится комбинация горячих клавиш, которая позволяет значительно ускорить работу в системе. На данный момент в программе CAMtastic! 2000 Designers' Edition доступно почти 200 макросов, и только 50 из них имеют назначенные по умолчанию комбинации горячих клавиш. Пользователь имеет возможность изменить эти настройки и назначить горячие клавиши наиболее часто выполняемым операциям.

Команда **Help » Shortcut Keys** вызывает диалоговое окно, показанное на рис 7.8. В списке слева выбирается необходимый макрос. Его краткое описание будет приведено в поле **Description**. Если данному макросу уже назначена какая-либо комбинация клавиш, то она будет показана в поле **Assigned Shortcuts**. Существующую комбинацию клавиш можно удалить с помощью кнопки **Remove**.

Назначение новой комбинации клавиш выполняется в диалоговом окне **Assign Shortcut**, вызываемом кнопкой **Create Shortcut**. Следует ввести новую комбинацию клавиш, после чего она запишется в текстовом окне. Если введенная комбинация правильна, необходимо нажать на кнопку **OK**, и для данного макроса в окне **Shortcut Keys** будет показана новая комбинация клавиш.

Каждому макросу может быть назначено несколько горячих клавиш, однако каждой комбинации горячих клавиш должен соответствовать только один макрос. Если новая введенная комбинация клавиш уже используется, появится сообщение, которое покажет, с каким макросом связана эта комбинация клавиш. Например, при попытке назначить какому-либо макросу горячей клавиши **T** появится сообщение о том, что эта клавиша уже используется для вызова макроса **View:Translucent**. Если в этом случае нажать кнопку **OK**, то новая настройка заменит старую.

При использовании горячих клавиш программа CAMtastic не различает регистр символов, поэтому одной букве может быть назначено только одно действие.

Для сброса всех пользовательских настроек горячих клавиш следует нажать кнопку **Reset All**.

Просмотр проекта

Программа CAMtastic! 2000 Designers' Edition позволяет открывать и просматривать до четырех проектов одновременно. Список всех открытых проектов приводится в

нижней части раздела меню Window с пометкой напротив активного документа. Каждый проект открывается в отдельном окне. Окна просмотра проекта могут быть упорядочены в границах рабочей области каскадом (команда **Window** » **Cascade**) и мозаикой (команда **Window** » **Tile**).

Управление окнами просмотра проекта осуществляется аналогично программам Windows. Двойной щелчок левой кнопкой мыши на заголовке разворачивает окно на весь экран. Свернутые окна могут располагаться в любой части окна программы, команда **Window** » **Arrange Icons** позволяет упорядочить их в нижней части окна.

В каждый конкретный момент времени может быть активен только один проект — его заголовок отличается от заголовков других проектов цветом, в зависимости от настроек цветовой палитры операционной системы.

Программа SAMtastic предлагает пользователю несколько различных инструментов масштабирования. Они могут быть доступны через комбинации горячих клавиш, кнопки на соответствующей панели инструментов и команды подменю **View** » **Zoom**. Для большинства команд масштабирования положение указателя мыши расценивается как центральная точка.

Рассмотрим основные инструменты масштабирования.

Zoom Window All: Показывает все объекты, содержащиеся в активном слое, центрируя их изображение на экране.

Zoom Previous: Возвращает предыдущий вид (функция похожа на Undo, но работает только для изменения: показывает увеличенный вид части текущего проекта, выделенной окном охвата).

Zoom In: Увеличивает текущее изображение.

Zoom Out: Уменьшает текущее изображение.

Dynamic Zoom: Позволяет динамически масштабировать окно посредством перемещения указателя мыши вверх или вниз по экрану.

Zoom Film Box: Показывает все объекты, которые находятся в текущем окне пленки. Эта команда доступна только тогда, когда включено отображение границ окна пленки с помощью команды **View** » **View Film Box**.

Zoom Dcode: Масштабирует изображение так, чтобы были оптимально видны все объекты, нарисованные с использованием активного инструмента (D-кода), причем сами объекты подсвечиваются.

Панорамирование изображения может быть выполнено клавишами управлением курсором. Также доступны две дополнительные опции панорамирования: динамическое и с указанием центра. Все эти опции вызываются с помощью комбинаций горячих клавиш, кнопок на соответствующей панели инструментов или команд подменю **View** » **Pan**. Для большинства команд панорамирования положение указателя мыши расценивается как центральная точка.

Center Pan: Вид изменяется таким образом, чтобы точка, где был выполнен щелчок левой кнопкой мыши, оказалась в центре экрана.

Dynamic Pan: Позволяет контролировать панорамирование движением указателя мыши при нажатой левой кнопке.

Выделение объектов

Следует отметить, что в большинстве Windows-программ пользователь сначала выделяет объекты, а потом указывает, что с ними делать. В программе SAMtastic используется противоположный подход. Здесь сначала предлагается выбрать действие (**Query**, **Chamfer** и т.д.), а потом указать объект, который следует обработать. Поясню иначе: если выбранные объекты являются частью действия, то инициализация действия автоматически переключит систему в режим выбора.

Выделения в программе являются кумулятивными, это означает, что при выделении нового объекта не отменяется предыдущее выделение. Объекты могут быть добавлены к выделению или удалены из него отдельно или группами.

Режим выделения будет оставаться активным до тех пор, пока действие не будет завершено (щелчок правой кнопкой мыши) или отменено (нажатие клавиши **Esc**).

При запуске функции, требующей выделения, указатель мыши изменит свой вид и будет показан в виде квадрата. В строке состояния будет выведено название выполняемой функции, а также инструкция по выделению объектов.

Выделение объекта происходит щелчком на нем левой кнопкой мыши, при этом он подсвечивается белым цветом. Щелчок левой кнопкой мыши на пустом месте переключает систему в режим задания окна охвата, о чем сигнализирует изменение формы указателя мыши на перекрестие. Если это не то, что вам нужно, просто щелкните левой кнопкой мыши в пустом месте, максимально приближенном к начальной точке, что равносильно указанию очень маленького окна охвата, и система вернется в режим выделения отдельных объектов.

Выделение окном охвата в программе SAMtastic! 2000 Designers' Edition имеет два различных режима: стандартный, когда выделяются все объекты, целиком попавшие внутрь окна охвата, и дополнительный, когда выделяются все объекты пересекающие или касающиеся окна охвата.

Стандартное окно выделения имеет рамку в виде сплошной линии. При выделении пересекаемых объектов границы окна охвата отображаются пунктирной линией.

Активизация любого из этих двух режимов производится нажатием на соответствующую кнопку, расположенную в панели инструментов **Select**. После указания окна охвата система возвращается в режим выделения отдельных объектов.

Существует другой способ выбора режима выделения окном охвата. Если после указания начальной точки окна сдвинуть мышью вправо, система автоматически перейдет в стандартный режим выделения. Если после указания начальной точки окна сдвинуть мышью влево, система перейдет в режим выделения пересекаемых объектов, о чем будет сигнализировать пунктирный тип линии отображения границы окна.

По умолчанию, для средств выделения доступны как линии, так и флэши, это означает, что пользователь может выделить любые объекты. Если требуется выбрать только те объекты, которые нарисованы флэшами, необходимо нажать на кнопку **Select**

Draws для ее отключения (если она была включена). Те же действия, но с кнопкой **Select Flash**, следует выполнять при необходимости выделения всех объектов, нарисованных линиями.

Выделение абсолютно всех объектов, нарисованных флэшами, линиями или и флэшами и линиями осуществляется нажатием на кнопку **All**.

То же действие, но только для объектов в отображаемой части проекта, выполняется с помощью кнопки **Display**.

Кнопка **Filter** на панели инструментов выделения предлагает пользователю дополнительные возможности выделения объектов. При нажатии на эту кнопку откроется показанное на рис 7.9 диалоговое окно **Selection Filter**.

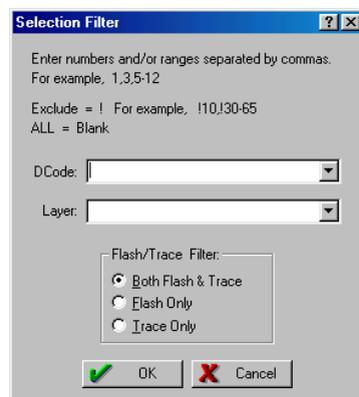


Рис. 7.9. Диалоговое окно Selection Filter.

D-коды и номера слоев могут быть перечислены отдельно или диапазоном с разделителем в виде запятой. Например, запись 1,3,5-7 будет означать, что в выделение включаются D-коды D1, D3, D5, D6 и D7 или слои с номерами 1, 3, 5, 6 и 7. Если строка останется пустой, то при выделении будут использоваться все D-коды или слои. Наличие восклицательного знака перед номером D-кода, слоя или диапазона означает исключение его из выделения.

Выпадающие списки заполняются по мере введения различных диапазонов, что упрощает последующее их использование.

Завершение режима выделения производится нажатием правой кнопки мыши. Отмена выполняемого действия в целом, производится нажатием **Esc**.

При ошибочном выделении объекта можно полностью прервать выполняемое действие нажатием клавиши **Esc**, или снять выделение с конкретных объектов кнопкой **Remove**. В этом случае будут работать все упомянутые методы выделения. Для возвращения в режим добавления к выделению следует нажать кнопку **Add**.

Наконец, программа CAMtastic запоминает все смешанные выделения и позволяет вам вернуться к предыдущему выделению после нажатия на кнопку **Previous Selection**.

Так как выделение играет ключевую роль в процессе работы над проектом, с каждой кнопкой на панели инструментов выделения связана определенная комбинация горячих клавиш.

Работа с проектами

При запуске программы CAMtastic! 2000 Designers' Edition автоматически создается новый пустой проект, который можно отредактировать, сохранить и закрыть. Имеется возможность импорта в проект файлов, созданных в других системах проектирования, печати и экспорта данных в различных форматах.

Создание нового проекта выполняется с помощью команды **File** » **New**. До первого сохранения данный проект не имеет имени, о чем свидетельствует надпись **Untitled** в заголовке окна проекта.

Команда **File** » **Open** позволяет открывать существующие проекты. Окно выбора файла предоставляет возможность предварительного просмотра содержимого выбранного САМ-файла до его открытия. Проекты, созданные предыдущими версиями программы SAMtastic будут автоматически распознаны, но функция предварительного просмотра для них будет недоступна.

Закрывать активный проект можно с помощью команды **File** » **Close**. Если в проекте были сделаны изменения, то система предложит предварительно сохранить проект.

Сохранение активного проекта выполняется командой **File** » **Save**. Если данный файл уже был сохранен ранее, то все текущие данные заменят старые. Если сохранение производится впервые, то появится диалоговое окно выбора имени и местоположения файла. Одновременно с топологической информацией в проекте будут сохранены все специфические настройки, которые будут восстановлены при последующем открытии программой SAMtastic. Команда **File** » **Save As** позволяет сохранить данный проект под другим именем.

Импорт файлов

Программа SAMtastic понимает большое количество форматов файлов, созданных в других системах проектирования. Основные опции, а также настройки по умолчанию задаются в диалоговом окне **Import/Export Preferences**, вызываемом двойным щелчком левой кнопки мыши на строке **Import/Export** в папке **Preferences**, расположенной на главной вкладке **Main** панели управления **Project Workspace**.

Настройки в этом диалоговом окне предназначены исключительно для работы с файлами в формате Gerber, однако одна из них является стандартной для всех модулей импорта программы SAMtastic. Нажатием кнопки **Import Settings** вызывается показанное на рис. 7.10 окно **Settings**.

В поле **Digits** задается количество цифр до и после десятичной точки (текстовые окна **Integer** и **Decimal** соответственно). Поле **Units** служит для переключения между дюймовой и метрической системами измерения. Поле **Type** предназначено для задания способа описания координат (абсолютная или относительная форма). Абсолютные координаты всегда отсчитываются от начала координат (**Origin**), относительные — от предыдущей точки. Поле **Zero Suppression** управляет режимом подавления незначащих нулей (**Leading** — в начале целой части, **Trailing** — в конце дробной части, **None** — нет подавления).



Рис. 7.10. Настройка функций импорта Gerber файлов.

Изменение настроек в этом окне делает их настройками по умолчанию для всех будущих сеансов импорта Gerber-файлов. Аналогичные окна настроек имеются у модулей импорта и экспорта в форматах Gerber, Drill, Mill и Rout.

Если все настройки заданы правильно, можно приступить к импорту внешних файлов. Он выполняется с помощью команды **File » Import**, в подменю которой имеются следующие опции: **Drill**, **DXF/DWG**, **Gerber**, **HPGL/HPGL2**, **Mill/Rout**. Здесь же присутствуют команды импорта стандартных таблиц апертур и пользовательских библиотек апертур.

Импорт NC Drill файлов

Файлы Drill содержат номера инструментов сверления, а также координаты всех X и Y отверстий в плате. Для просмотра Drill-файлов используется команда **File » Import » Drill**.

Хотя программа CAMtastic может интерпретировать широкий набор форматов файлов сверления, на рис. 7.11 приведен наиболее предпочтительный формат. После выбора файла появится диалоговое окно **Import Drill Data**. Программа в качестве файлов сверления по умолчанию ищет файлы с расширениями **.DRL** и **.DPT**, так как они являются стандартными. Тем не менее, может быть импортирован любой ASCII файл с произвольным расширением. Нажатие кнопки **Units** вызывает стандартное диалоговое окно **Settings**, описанное ранее.

В некоторых случаях файлы сверления могут не содержать описание диаметров инструментов, тогда таблицы инструментов загружаются отдельно. Если описание сверла не обнаружено, программа будет использовать значение, приведенное в поле **Shape/Default Hole Size** диалогового окна **Import Drill Data**. Кнопка **Tool Table** служит для вызова окна назначения существующих инструментов вручную, их модификации и импортирования новых элементов. Если импортированные описания инструментов имеют номера, совпадающие с уже существующими, им будут автоматически назначены новые номера.

Импорт DXF и DWG файлов

Файлы системы AutoCAD могут быть импортированы в программу CAMtastic! 2000 Designers' Edition напрямую. Команда **File » Import » DXF/DWG** вызывает мастер импорта DXF/DWG файлов (рис. 7.12).



Рис. 7.11. Структура формата файлов NC Drill.

Мастер импорта имеет широкий набор настроек, для облегчения работы с которыми введены четыре заранее заданных набора в зависимости от назначения платы: **Standard PCB**, **Microwave RF**, **Chemical Drawings** и **Fab Drawings**.

Если пользователь ошибочно выберет такой масштаб, что размеры платы превысят допустимые, в дальнейшем у него будет возможность его изменить.

В выпадающем списке **Units** следует выбрать единицы измерения, используемые в импортируемом файле. Опция **Fill TrueType Fonts** управляет заливкой текстовых надписей, выполненных TrueType шрифтами.



Рис. 7.12. Мастер импорта DXF/DWG файлов.

Если нет мастера, правильно читающего импортируемые файлы, пользователь имеет возможность определить индивидуальные характеристики, выбрав опцию **Skip Wizard** и нажав **OK**. Откроется показанное на рис. 7.13 диалоговое окно **Import DXF/DWG**. Каждый обнаруженный слой будет автоматически включен для импорта. Для слоя доступны четыре столбца опций.

Опция **FCB (Fill Closed Boundaries)**: Если эта опция включена, области, ограниченные замкнутыми полилиниями нулевой толщины или сплошными линиями, будут автоматически залиты. Если эта опция выключена, линии будут прорисованы как контуры, а их толщина будет установлена из параметров диалогового окна **Text Options**.

Опция **OWP (Outline Width Polylines)**: Если эта опция включена, то полилинии будут импортированы в точном соответствии с проектом AutoCAD. Если эта опция выключена, различные сегменты полилиний будут иметь одинаковую ширину, равную начальной.

Опция **FRD (Flash Round/Donut Pads)**: Если эта опция включена, то всем круглым объектам в DXF/DWG файле будет назначена соответствующая апертюра, а сами объекты будут прорисованы флешами. Если эта опция выключена, такие объекты будут прорисованы как контуры, а их толщина будет определяться из установок окна **Text Options**.

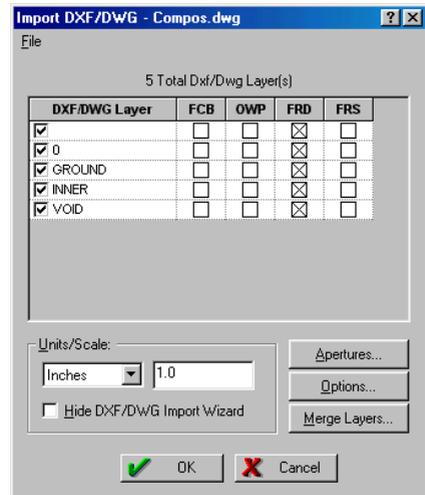


Рис. 7.13. Пользовательские настройки импорта DXF/DWG файлов.

Опция **FRS (Flash Rectangle/Square Pads)**: Если эта опция включена, то всем квадратным и прямоугольным объектам в DXF/DWG файле будет назначена соответствующая апертюра, а сами объекты будут прорисованы флешами. Если эта опция выключена, такие объекты будут прорисованы как контуры, а их толщина будет определяться установками диалогового окна **Text Options**.

В этом же диалоговом окне имеется собственное меню **File**, которое позволяет сохранять и загружать файлы настроек модуля импорта DXF/DWG файлов или таблицы апертур. Здесь же можно изменить масштаб проекта, задав нужное значение в поле **Scale**. Включение опции **Hide Import Wizard** позволяет отказаться от вызова окна выбора мастера импорта.

Если пользователь знает формат DXF/DWG не слишком хорошо, рекомендуется модифицировать только основные настройки, находящиеся в диалоговом окне **Import DXF/DWG**. После загрузки внешнего DXF/DWG файла в проект программы CAMtastic его содержание можно без труда изменить.

В этом диалоге доступны три дополнительные кнопки. Кнопка **Merge Layers** позволяет настроить слияние слоев. Для импорта DXF/DWG файлов со слиянием слоев необходимо наличие пустого проекта, в любом случае специальное предупреждение дает возможность пользователю отменить это действие перед перезаписью любых существующих данных. Кнопка **Aperture** вызовет диалоговое окно **Aperture Table**. Кнопка **Options** является уникальной для модуля импорта DXF/DWG файлов. Ее нажатие вызывает показанное ниже диалоговое окно **Advanced Options** (рис. 7.14).

Данное диалоговое окно позволяет полностью контролировать импорт DXF/DWG файла в проект программы CAMtastic. Здесь можно выбрать один из трех OWP-режимов: сегментирование, квадратные углы или скошенные углы. В текстовых полях здесь можно задать ширину линий (**Line Width**) и коэффициент заполнения прямоугольников (**Rectangle Flash Ratio**). Остальные опции позволяют вводить заливку и оконечные элементы полилиний, игнорировать штриховку, перемещать блоки из слоя 0 в назначенный слой.

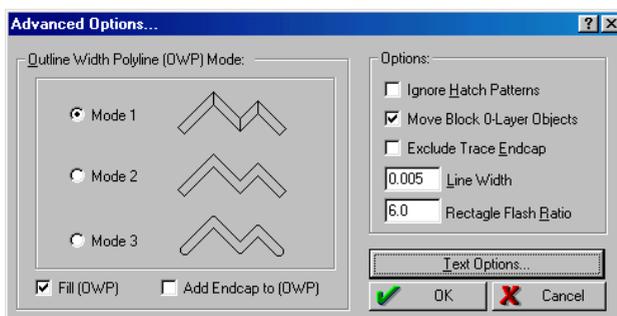


Рис. 7.14. Пользовательские настройки импорта DXF/DWG файлов.

Когда большинство конструкторов в AutoCAD создают библиотеки блоков, все объекты блока обычно реализуются в слое 0. Впоследствии эти блоки вставляются в другие слои, слои переназначаются, но внутренним образом они по-прежнему остаются в слое 0. Если эта опция включена, программа CAMtastic будет перемещать все объекты блока в нужный слой, аналогично тому, как это делает AutoCAD. Если опция выключена, все объекты блоков будут записываться в слой 0.

Кнопка **Text Options** запускает диалоговое окно, управляющее процессом импорта текст из DXF/DWG файлов (рис. 7.15).

Use AutoCAD SHX: Позволяет использовать для импортирования текстов стандартные для AutoCAD файлы шрифтов.SHX.

Use TrueType: Позволяет использовать шрифт TrueType, который задается нажатием на кнопку Select.

Select TrueType Font Quality: Задает качество прорисовки шрифтов TrueType.

Use Polygon/Raster Fill on TrueType: Заполняет шрифты TrueType растровой заливкой, что предпочтительнее, чем векторная заливка, так как в последнем случае увеличивается размер файла.

Use Same Line Width for All: Непосредственно задает ширину линий для прорисовки всех текстовых надписей.

Using Ratio (Height/Line Width): Задает отношение высоты букв и ширины линий для прорисовки всех текстовых надписей.

Импорт Gerber файлов

Файлы в формате Gerber могут быть загружены в проект как группой, так и по отдельности. Если на диске имеется папка, содержащая несколько Gerber-файлов, которые должны быть импортированы в проект одновременно, рекомендуется использовать функцию **Gerber Quick Load**. Если необходимо выполнить выборочную загрузку файлов, следует пользоваться стандартным мастером импорта Gerber-файлов.

Функция быстрой загрузки (**Gerber Quick Load**) позволяет автоматически распознать и загрузить все файлы в формате Gerber и файлы апертур в указанной папке.

Для открытия диалогового окна **Gerber Quick Load** необходимо выполнить команду меню **File » Import » Gerber Quick Load** (рис. 7.16).

Здесь следует указать диск и папку, содержащую файлы в формате Gerber. Программа CAMtastic автоматически определит необходимые для импорта настройки, тем не

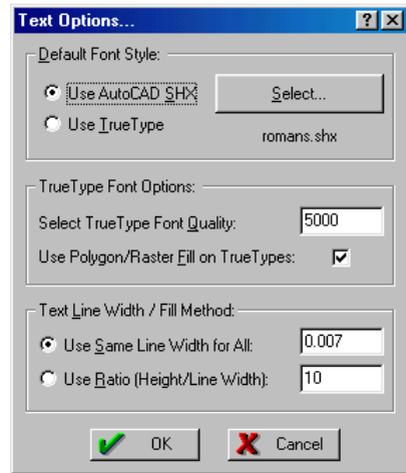


Рис. 7.15. Настройки текста при импорте DXF/DWG файлов.

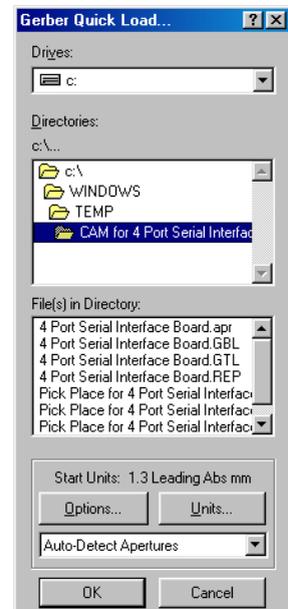


Рис. 7.16. Быстрая загрузка Gerber файлов.

менее, пользователь имеет возможность изменить их по своему усмотрению.

Чтобы максимальным образом использовать возможности этой функции, перед выполнением импорта следует убедиться, что все необходимые Gerber-файлы хранятся в одной папке. Для открытия описанного ранее диалогового окна **Settings** используется кнопка **Units**.

С помощью кнопки **Options** вызывается показанное на рис. 7.17 диалоговое окно **Import Gerber Options**.

Здесь имеются три опции:

Use 360 Degree Arcs as default: если эта опция выбрана, программа CAMtastic позволит импортировать дуги, превышающие 90 градусов.

End Gerber Block at NewLine: устанавливает режим, когда аппаратный возврат строки будет восприниматься как конец блока данных.

End of Gerber Block Character: задает символ, который будет восприниматься как конец блока данных.

Настройки по умолчанию диалоговых окон **Import Gerber Options** и **Settings** могут быть введены в **Import/Export Preferences**.

В выпадающем списке **Aperture Wizard Rule** задается шаблон для импорта таблиц апертур. Если неизвестно, в какой системе САПР были сгенерированы данные Gerber-файлы, то следует использовать элемент списка **Auto-Detect Apertures**, который всегда предлагается по умолчанию. Имеется возможность описания собственных пользовательских шаблонов импорта апертур.

Стандартная функция импорта Gerber-файлов в любом из поддерживаемых форматов (RS-274D, RS-274X, File9000) вызывается командой меню **File » Import » Gerber**. Далее необходимо выбрать один или несколько (удерживая клавиши **SHIFT** и **CTRL**) Gerber-файлов и нажать кнопку **Open**, после чего появится окно **Import Gerber Files**, показанное на рис. 7.18.

Некоторые из расположенных здесь органов управления были описаны ранее.

- Кнопка **Settings** открывает диалоговое окно **Settings**.
- Кнопка **Load Options** открывает диалоговое окно **Import Gerber Options**.
- Кнопка **Aperture Wizard Rule** включает список правил быстрой загрузки.

Кроме них в этом окне имеются три дополнительные опции.

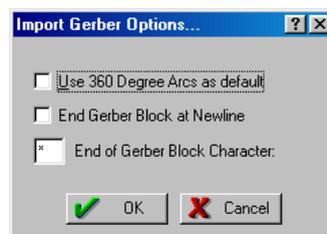


Рис. 7.17. Опции импорта Gerber файлов.



Рис. 7.18. Стандартная загрузка Gerber файлов.

- **Gerber Version:** переключатель между форматами **Gerber RS-274** (подразумевается RS-274D), **Gerber RS-274X** и **Fire9000**.
- **Auto-Detect Gerber Formats:** включает режим автоматического определения всех опций и настроек, необходимых для корректной загрузки Gerber файла.

Для импорта файла в формате Gerber RS-274D необходима внешняя таблица апертур, в то время как в форматах RS-274X и Fire9000 вся информация об апертурах содержится внутри самого файла.

- **Aperture List:** позволяет указать таблицу апертур для импортируемого файла.

Импорт файлов в формате HPGL/HPGL2

Программа CAMtastic! 2000 Designers' Edition позволяет импортировать файлы в формате HPGL (*.HPG, *.HP и *.PLT). Выбор одного или нескольких (удерживая клавиши **SHIFT** и **CTRL**) файлов выполняется в окне, вызываемом с помощью команды меню **File » Import » HPGL/HPGL2** (рис. 7.19).

Если топология в файле прорисована с использованием нескольких различных инструментов, программа CAMtastic может выполнить разделение информации по слоям (каждому инструменту соответствует свой слой) или апертурам (каждому инструменту назначается отдельный D-код). Пользователю следует выбрать необходимый способ в поле **Pen Separation**, если разделение не требуется, то выбрать опцию **None** и задать толщину инструмента по умолчанию в поле **Pen Width**.

Если у вас есть сложные шрифты, используемые в системе AutoCAD, их можно распечатать в файл в формате HPGL и затем загрузить в программу CAMtastic.

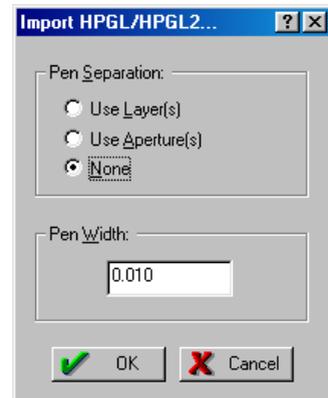


Рис. 7.19. Импорт HPGL/HPGL2 файлов.

Импорт файлов в формате NC Mill/Route

CAMtastic! 2000 Designers' Edition позволяет импортировать управляющие файлы для фрезеровальных станков в формате NC Mill/Route (*.RTE, *.ROU). Выбор одного или нескольких (удерживая клавиши **SHIFT** и **CTRL**) файлов выполняется в окне, вызываемом с помощью команды меню **File » Import » Mill/Route** (рис. 7.20).

Кнопка **Units** вызывает диалоговое окно **Settings**. Кнопка **Mill Table** служит для назначения инструментов

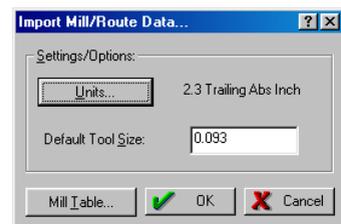


Рис. 7.20. Импорт NC Mill/Route файлов.

в таблице **Tool Table**. В противном случае для всех линий будет использован инструмент с диаметром, заданным в поле **Default Tool Size**.

Импорт файлов апертур

В некоторых случаях, когда CAMtastic не может автоматически определить необходимый для импорта Gerber-файла шаблон таблицы апертур, пользователь может принудительно назначить для него нужный шаблон с помощью команды меню **File » Import » Aperture File (using Wizard Formats)** (рис. 7.21).

В появившемся окне **Import Aperture File** имеется выпадающий список **Existing Wizard Formats**, в котором перечислены все шаблоны, входящие комплект стандартной поставки программы CAMtastic, а также дополнительные, созданные пользователем шаблоны.

После выбора нужного мастера следует задать местоположение файла апертур в **Aperture File**. Для поиска файла на жестком диске необходимо использовать кнопку, расположенную справа от текстового поля. После нажатия кнопки ОК система выдаст запрос, что делать с загружаемой таблицей апертур: полностью обновить существующий список апертур, или добавить апертуры из файла к существующему списку.

С помощью команды **File » Import » Custom Aperture Library File** можно выполнить импорт пользовательских библиотек апертур. Использование библиотек апертур позволяет значительно упростить работу с пользовательскими апертурами сложной формы.

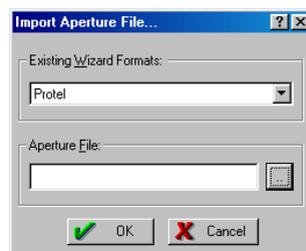


Рис. 7.21. Импорт файлов апертур.

Экспорт файлов

CAMtastic! 2000 Designers' Edition может экспортировать данные во всех форматах, которые импортирует. Настройки экспорта по умолчанию, включая оптимизацию экспортируемых файлов, задаются в диалоговом окне **Import/Export Preferences**.

Экспорт файлов NC Drill

Функция экспорта позволяет выполнить экспорт простых файлов в формате NC Drill из текущего проекта. Для этого необходимо выполнить команду меню **File » Export » Drill Data**, после чего на экране

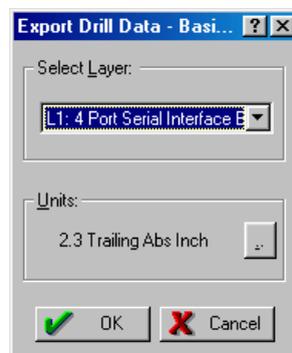


Рис. 7.22. Экспорт Drill файлов.

появится показанное на рис. 7.22 диалоговое окно **Export Drill Data — Basic**.

Здесь нужно выбрать слой, который будет экспортироваться и при необходимости изменить единицы измерения и формат численных данных.

После нажатия кнопки **OK** пользователю будет предложено задать местоположение и имя генерируемого файла. Однако вместо одного в заданной папке будут созданы два файла с одинаковым именем, но разными расширениями **.DRL** и **.RPT**. Файл с расширением **.DRL** содержит управляющие инструкции в формате Excellon. Этот файл может быть импортирован в другие проекты CAMtastic. Файл с расширением **.RPT** является отчетом, содержащим список используемых инструментов и количество отверстий, изготавливаемых с помощью каждого инструмента.

Экспорт DXF файлов

Для экспорта топологии в формате DXF используется команда меню **File » Export » DXF**, вызывающая показанное на рис. 7.23 диалоговое окно **Export DXF**.

Все обнаруженные в проекте слои будут выделены автоматически. Щелчок левой кнопкой мыши по любому из них снимает выделение. Функция экспорта DXF файлов с возможностью выбора слоев незаменима, если требуется вывести содержимое только одного слоя из проекта печатной платы, так как встроенная в систему Protel 99 SE функция экспорта DXF файлов выводит информацию обо всех без исключения слоях, в результате чего полученный файл может иметь очень большой размер.

В поле **Output Control** задается способ экспорта линий: линии нулевой толщины **0-Width**, залитые без оконечных элементов **Filled (No Endcaps)** или залитые с оконечными элементами **Filled (End Caps)**. В последнем случае в системе AutoCAD линии будут выглядеть точно так же, как и в программе CAMtastic.

Для уменьшения размера файлов DXF имеется возможность использовать фильтр, блокирующий экспорт любых ненужных в системе AutoCAD D-кодов, например, используемых в программе CAMtastic для заполнения полигонов. Здесь же расположена опция, предписывающая конвертировать текстовые надписи в полилинии.

Как правило, большинство DXF файлов, полученных из Gerber-файлов, имеют очень большие размеры. Для уменьшения размера DXF файла рекомендуется использовать не залитые линии, а линии с нулевой толщиной. Это также позволит увеличить скорость обновления экрана, если DXF/DWG-файл используется в качестве эталонного.

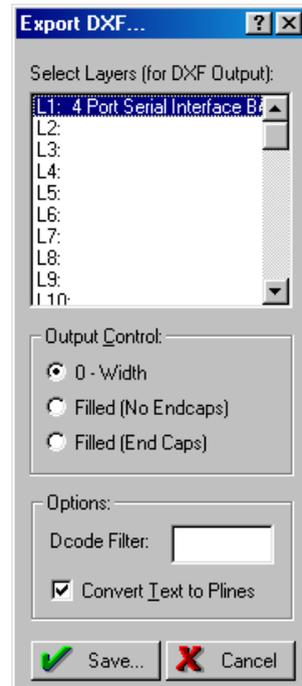


Рис. 7.23. Экспорт DXF файлов.

После нажатия на кнопку **Save** система предложит указать местоположение и имя генерируемого DXF файла.

Экспорт Gerber файлов

Для экспорта топологии в формате Gerber используется команда меню **File » Export » Gerber**, вызывающая показанное на рис. 7.24 диалоговое окно **Export Gerber Data**.

Левая кнопка в поле **Format** позволяет последовательно переключать выходной формат (**RS-274D**, **RS-274X** или **Fire9000**). Кнопка **Settings** вызывает описанное ранее стандартное диалоговое окно **Settings**.

В поле **Options** имеются четыре опции:

Опция **Use Arcs (G75)** включает передачу в генерируемый Gerber-файл истинных дуг. Разумеется, в проекте CAMtastic дуги должны присутствовать именно как дуги, а не как их аппроксимация. Если опция выключена, то в генерируемом файле дуги будут представлены в виде последовательности коротких линий.

Опция **Use Step & Repeat Codes (if any)** позволяет включить в генерируемый Gerber-файл коды Step & Repeat. Понятно, что в проекте CAMtastic эти коды уже должны присутствовать. Если опция выключена, коды Step & Repeat будут преобразованы в отдельные объекты.

Опция **Separate Composite Layers into individual files** управляет разделением композитных слоев в отдельные Gerber файлы. В проекте Camtastic информация о композициях уже должна присутствовать.

Опция **Convert Polygons to Vector Fill** позволяет выводить присутствующие в проекте полигоны в виде заштрихованных областей.

В поле **File** будут показаны все обнаруженные в проекте слои. Выключение флажка слева от каждого показанного слоя запретит генерацию Gerber-файла для данного слоя. Для выключения генерации абсолютно всех слоев используется кнопка **Clear All**, которая после нажатия преобразуется в кнопку **Write All**, а ее использование приведет к включению всех слоев. Каждому выходному файлу будет автоматически назначено уникальное имя, полученное из имени соответствующего слоя, которое в последствии может быть изменено. Если все файлы должны иметь одинаковое расширение, его следует ввести в текстовое поле, расположенное рядом с кнопкой **Apply Gerber File Extension**, и затем нажать на эту кнопку.

Имя папки, куда будут сохранены все генерируемые Gerber-файлы, задается в текстовом поле, расположенном рядом с кнопкой **Apply Directory**. Нажатием на кнопку

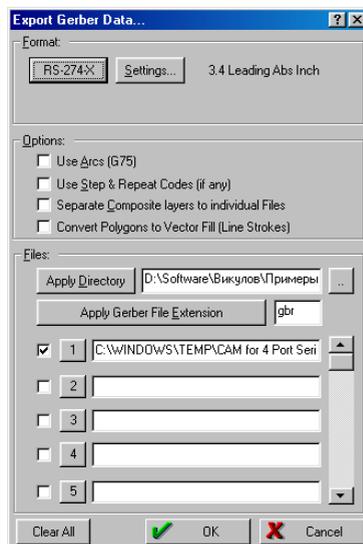


Рис. 7.24. Экспорт Gerber файлов.

добавляется адрес папки к именам файлов. При необходимости пользователь может изменить путь сохранения индивидуально для файлов. Щелчок левой кнопкой мыши по кнопке с номером слоя, расположенной слева от имени файла, запускает команду экспорта только данного слоя.

Экспорт файлов в формате NC Mill/Route

Экспорт файлов в формате NC Mill/Route выполняется с помощью команды **File » Export » Mill/Route**, вызывающей диалоговое окно **Export Mill/Route Data** (рис. 7.25).

Для загрузки ранее заготовленного и сохраненного в текстовом файле заголовка используется присутствующая здесь команда меню **Header File**.

Здесь из выпадающего списка **Select Layer** следует выбрать нужный слой или задать на экспорт все слои (опция **All Layers**) и, если необходимо, изменить заголовок файла.

Кнопка **Units** вызывает стандартное диалоговое окно **Settings**. Если в проекте присутствуют коды **Step & Repeat**, следует включить опцию **Use Step & Repeat Codes**. Кнопка **Tool Table** открывает таблицу инструментов.

Экспорт списка соединений

Экспорт сгенерированного в проекте списка соединений выполняется с помощью команды меню **File » Export » NetList**. Эта функция позволяет экспортировать простые списки соединений аналогичные файлам тестовых точек для тестирующего оборудования. Настройка формата списка соединений производится в показанном на рис. 7.26 диалоговом окне **Export NetList**.

В поле **Net Points** определяется, какие узлы включать в список соединений: все (**All**) или только конечные (**Ends Only**). Генератор автоматически определит координаты центральных точек всех контактных площадок и переходных отверстий. В поле **Pcb Side** указываются используемые стороны платы.

Для успешного выполнения операции экспорта



Рис. 7.25. Экспорт NC Mill/Route файлов.

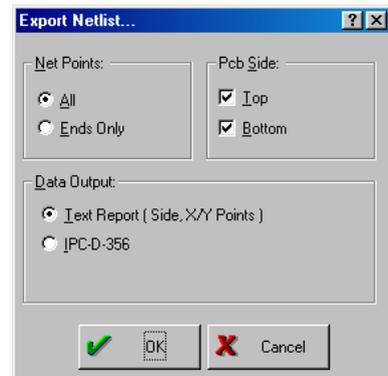


Рис. 7.26. Экспорт списка соединений.

списка соединений необходимо, чтобы каждая в таблице слоев (**Layer Table**) были правильно назначены слои **Top** и **Bottom**.

Список соединений может быть сохранен в текстовом формате или в формате IPC-D-356. Генерируемый при этом файл с расширением **.IPC** является основным способом обмена информацией об электрических соединениях между различными CAD/CAM системами.

Экспорт активного проекта в формате IPC-D-350 выполняется с помощью команды меню **File** » **Export** » **IPC-D-350**. Здесь нет никаких настроек, поэтому от пользователя требуется указать только местоположение генерируемого IPC-файла на диске.

Экспорт BMP файлов

Программа CAMtastic! 2000 Designers' Edition позволяет сохранить любой участок топологии в виде растрового файла в формате BMP. Настройки этой процедуры выполняются в диалоговом окне **Export Bitmap**, вызываемом командой меню **File** » **Export** » **Bitmap** (рис. 7.27).

Растровое изображение может быть создано с различным разрешением, которое задается в выпадающем списке **DPI (Dots per Inch)**. Здесь также задается цветной или монохромный режим вывода. Отметим, что генерируемый в растровом файле фон всегда белый.

Программа CAMtastic имеет довольно широкий набор опций печати. Например, имеется возможность предварительного просмотра распечатки и копирования текущего вида в буфер обмена системы Windows, откуда он может быть вставлен в любое другое приложение Windows и распечатан его средствами.

Несмотря на то, что в программе CAMtastic топология отображается светлыми линиями на темном фоне, при печати на цветных или монохромных принтерах она будет показана темными линиями на белом фоне.

Вывод файлов на печать

Печать в программе CAMtastic! 2000 Designers' Edition выполняется по команде меню **File** » **Print**. При этом выводится показанное на рис. 7.28 диалоговое окно **Print Setup/Preferences**.

Здесь имеются следующие настройки:

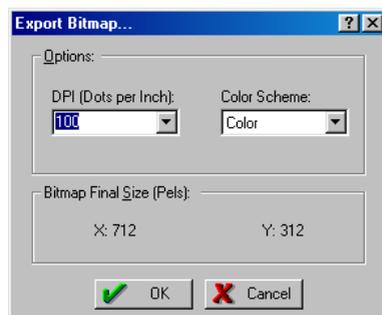


Рис. 7.27. Экспорт BMP файла.

Print Area: Переключатель режима печати текущего вида (**Display**) и всего изображения (**Extents**).

Print Color: Переключатель режима монохромной (**Black & White**) и цветной печати (**Color**).

Print Scaling: Переключатель режима масштабирования с подгонкой под размер листа (**Fit to Page**) и с фиксированным масштабным коэффициентом (**User Scale**).

Print Options: Предлагает три опции: центрировать изображение на странице (**Center Image on Page**), печатать информационный заголовок (**Print Page Info Header**), печатать каждый слой на отдельной странице (**Separate Page for each Layer**).

Printer Setup: Эта кнопка вызывает стандартное диалоговое окно настройки принтера, включающее выбор принтера, задание его свойств, ориентацию страницы, а также источник и размер бумаги. Кнопка **Network** позволяет использовать для печати сетевой принтер.

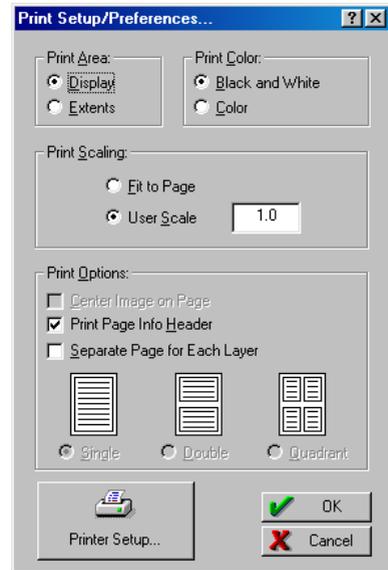


Рис. 7.28. Настройка печати.

Быстро распечатать видимую область изображения можно, выполнив команду меню **File » Print Screen**.

Если вы не хотите изменить настройки, нажмите кнопку **OK**.

При нажатии на кнопку **Printer Setup** откроется дополнительное диалоговое окно, в котором программа CAMtastic автоматически определит установленные в операционной системе принтеры и предложит стандартный набор опций настройки, зависящих от выбранного принтера. Для поиска сетевого принтера необходимо нажать кнопку **Network**. Настройка параметров принтера производится после нажатия на кнопку **Properties**.

После нажатие кнопки **OK** в диалоговом окне **Print Setup/Preferences**, появится последнее окно, где можно указать количество требуемых копий. Для отправки чертежа на печать следует нажать кнопку **OK**.

Для предварительного просмотра изображения перед печатью используется команда меню **File » Print Preview**. Появится описанное выше окно **Print Setup/Preferences**, в котором следует сделать все необходимые настройки и нажать кнопку **OK**. Откроется окно предварительного просмотра распечатки с кнопками масштабирования и прокрутки страниц. Нажатием кнопки **Print** вызывается описанное выше диалоговое окно **Print**, отправляющее документ на печать.

Работа с буфером обмена

При копировании в буфер обмена изображение будет выглядеть точно так же, как оно отображается в программе CAMtastic с темным фоном и видимыми точками сетки.

Если необходимо распечатать топологию из другого приложения Windows, можно выполнить ее экспорт в различные форматы, включая растровое изображение. Кроме того, текущий вид окна проекта может быть скопирован в буфер обмена системы Windows, для этого используется команда меню **File » Copy to Clipboard**, после чего вставлен в другое приложение и распечатан его средствами.

Работа в программе CAMtastic

Рассмотрим ряд основных понятий программы CAMtastic! 2000 Designers' Edition, такие как D-коды (апертуры), слои, режимы рисования, инструменты для сверления и фрезерования.

Апертуры

Под коды апертур различных размеров и форм в проектах CAMtastic отведены D-коды из диапазона от D10 до D999. Список используемых D-кодов называется таблицей апертур, для редактирования которой предназначен модуль **Aperture Table**. При импорте топологии в программу модуль **Aperture Table** автоматически генерирует все D-коды, необходимые для получения корректной топологии.

Панель инструментов **Aperture** содержит кнопку вызова модуля **Aperture Table** и выпадающий список всех определенных в проекте D-кодов (рис. 7.29).

В обычном состоянии в выпадающем списке D-кодов отображается текущий код, который используется при любых операциях рисования. Чтобы сделать текущим другой D-код, необходимо выбрать его в выпадающем списке.

Форма апертуры, соответствующей каждому D-коду должна быть описана в модуле **Aperture Table**, однако, это можно сделать, когда на топологии уже имеются объекты, прорисованные с помощью данной апертуры. Однажды описанные форма или размер апертуры могут быть изменены в любой момент работы над проектом. Любые изменения в существующих D-кодах автоматически приводят к соответствующему изменению в графических объектах, при прорисовке которых они использовались.



Рис. 7.29. Список используемых в проекте D-кодов.

Модуль **Aperture Table** можно запустить с вкладки **Main View** панели управления **Project Workspace** или с помощью кнопки **Aperture Table** на панели инструментов **Aperture**.

В новом проекте этот список апертур будет пустым до тех пор, пока пользователь не импортирует в проект какую-либо топологию или не определит D-коды вручную. Программа CAMtastic автоматически создает список апертур из импортируемых файлов, например, Gerber-файлов. Ручное определение размеров и форм производится в диалоговом окне Aperture (рис. 7.30).

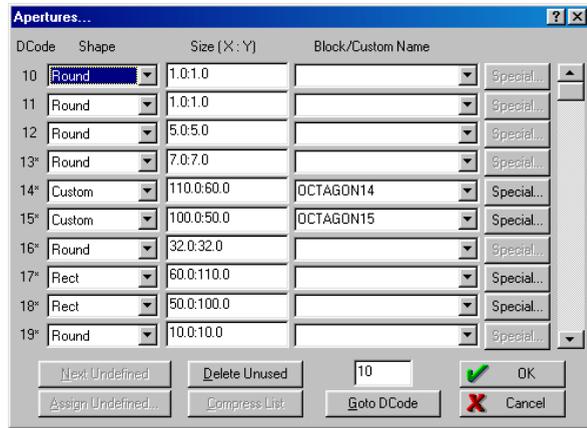


Рис. 7.30. Настройка апертур проекта.

Для описания апертур доступны D-коды с D10 до D999. Символ * (звездочка) рядом с номером D-кода говорит о том, что данная апертура используется в проекте. Для описания формы апертуры в выпадающем списке Shape доступно четырнадцать опций, параметры которых приведены в таблице 7.1, приведенной ниже. Формат описания размеров для различных форм отличается незначительно.

Таблица 7.1

Форма	Формат размера
Custom (пользовательская)	X:Y(размеры)
Diamond (ромб)	Длина:Высота
Donut (кольцо)	Внешний диаметр:Внутренний диаметр
Oblong (прямоугольник)	Длина:Высота
Octagon (восьмиугольник)	Длина:Высота
Round (круг)	Диаметр
Polygon (многоугольник)	Длина
Polygon Void (многоугольный вырез)	Длина
Square (квадрат)	Длина
Rect (прямоугольник)	Длина:Высота
RoundRect (скругленный прямоугольник)	Длина:Высота

Target (мишень)	Внешний диаметр
Thermal (тепловой барьер)	Внешний диаметр:Внутренний диаметр
Undefined (неопределенный)	Нет

Формы апертур типа **Custom** для более точного описания должны иметь специальные атрибуты. Например, если для новой апертуры просто выбрать форму **Custom** и правильно ввести величины X и Y, то по выходу из таблицы новая апертура создана не будет. С другой стороны, если в проекте уже используется некая апертура простой формы, то после изменения ее формы на **Custom** и выхода из модуля **Aperture Table**, все объекты будут прорисованы маленькими квадратами. При повторном запуске модуля **Aperture Table** нажатие на кнопку **Special** вызывает ошибку, так как пользователь пытается связать свои атрибуты с их текущей формой.

Угол поворота может быть произвольным в диапазоне от 0 до 360 градусов. Отрицательные значения не допускаются.

Правильный метод описания апертуры заключается в том, чтобы сначала выбрать форму, которая может иметь специальные атрибуты. Любую форму кроме **Round**, **Donut**, **Polygon**, **Polygon Void** можно вращать. Форма **RoundRect** позволяет задавать радиус угла закругления. Форма **Target** позволяют изменять число (от 1 до 4) и ширину колец. Форма **Thermal** позволяет изменять число (2 или 4) и ширину проводников.

Как только апертуре будут назначены специальные атрибуты, ее форма автоматически изменится на **Custom**. После этого пользователь может изменять любые специальные атрибуты, но не может изменить размеры, как это делается с простыми апертурами. Чтобы изменить размеры пользовательской апертуры, необходимо сначала изменить ее форму на простую, затем отредактировать размеры и задать специальные атрибуты.

При импорте топологий в формате Gerber, DXF или DWG в программе CAMtastic автоматически создается таблица апертур и в список **Block/Custom Names** заносятся все имена блоков и форм, присутствующие в оригинальном файле. Если пользователь хочет назначить какому-либо коду специфический блок, сначала надо определить форму и размеры апертуры, а затем выбрать нужный блок из списка **Block/Custom Names**. Дополнительные блоки включаются в этот список, если какая-либо пользовательская апертура использовалась в проекте для прорисовки объектов.

Если в текущем проекте используются неопределенные апертуры, кнопка **Next Undefined** в окне **Aperatures** будет активна. С ее помощью можно последовательно просматривать в списке все апертуры типа **Undefined**.

Рядом с номером используемого в проекте кода в таблице апертур ставится символ * (звездочка). С помощью кнопки **Delete Unused** можно найти все неиспользуемые в проекте D-коды и сбросить их установки в значения по умолчанию. Это бывает полезно, когда требуется освободить места для новых апертур, так как программа CAMtastic в одном проекте не может содержать более 990 уникальных D-кодов.

Для быстрого перехода к определенному D-коду используется кнопка **Goto Dcode**, номер кода вводится в расположенном рядом с ней текстовом поле.

Слои

Все изображения в САМtastic! 2000 Designers' Edition рисуются на определенном слое или импортируются в заданный слой. При создании нового проекта, чтобы в окне просмотра показалось какое-либо изображение, необходимо определить хотя бы один слой. Если пользователь не добавил слой вручную, слой будет создан автоматически при попытке нарисовать что-либо с помощью выбранного D-кода.

Программа САМtastic в одном проекте может хранить до 90 слоев. Для управления слоями используется опция **Layer View** на панели управления **Project Workspace**, панель инструментов **Layers** и модуль **Layer Table**.

Панель инструментов **Layers** почти всегда активна и предлагает несколько функций работы со слоями. Для включения отображения всех слоев используется кнопка **All ON**, для выключения — кнопка **All OFF**. Новый слой создается кнопкой **Add Layer**.

Как уже отмечалось ранее, текущий активный слой имеет крестик на цветном квадрате в панели управления **Project Workspace**. Имя активного слоя также отображается в выпадающем списке **Layer Control**, расположенном на панели инструментов **Coordinate** (рис. 7.31).

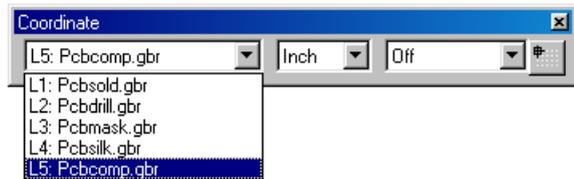


Рис. 7.31. Выбор активного слоя.

Для изменения текущего слоя следует выбрать из этого выпадающего списка другой слой. Каждый слой имеет свой уникальный номер, присвоенный в том порядке, в котором он был создан или импортирован.

Несколько слоев можно объединить различными способами. Слияние слоев будет означать добавление нового слоя всякий раз, когда для прорисовки сложных объектов (например, кольца) используются композитные слои.

Главным инструментом работы со слоями является таблица слоев (модуль **Layer Table**). Таблицу слоев можно открыть двумя способами: нажатием на соответствующую кнопку на панели инструментов **Layers** или двойным щелчком мыши по ана-

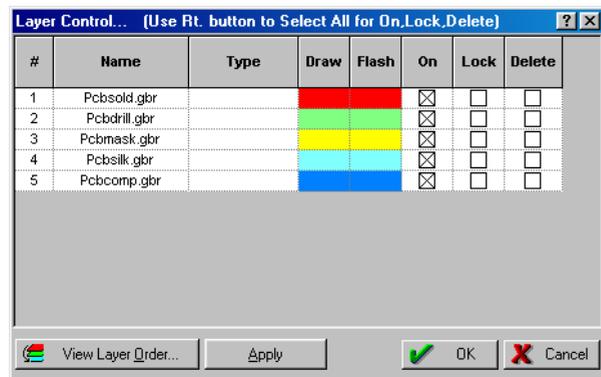


Рис. 7.32. Настройка слоев проекта.

логичной иконке на вкладке **Main View** панели управления **Project Workspace**.

Вид таблицы слоев показан на рис. 7.32. Здесь имеются восемь столбцов (**Layer Number, Name, Type, Draw Color, Flash Color, On, Lock** и **Delete**), в которых определяются соответствующие атрибуты слоев.

Layer Number: Показывает номер слоя. Каждый слой имеет номер, данный ему в том порядке, в котором он создавался или импортировался. Порядок номеров не может быть изменен, хотя порядок рисования слоев можно изменить, об этом будет сказано позже).

Name: Показывает имя слоя. Изменить имя слоя можно после щелчка левой кнопкой мыши на ячейке таблицы с его именем.

Type: Показывает тип слоя, если таковой определен. Некоторые функции программы CAMtastic для правильной интерпретации данных требуют задания типа слоя. После щелчка левой кнопкой мыши на ячейке из этого столбца, появится выпадающий список, из которого можно выбрать один из следующих типов: **Top, Bottom, Internal, Neg Plan, Pos Plan, Drill Top, Drill Bot, Drill Int, Silk Top, Silk Bot, Mask Top, Mask Bot, Paste Top, Paste Bot, Rout/Mill, Insulator, Refdes Top, Refdes Bot, Temporary, Border, Dark Image, Clear Image** или же не выбрать ничего.

Draw Color: Отображает цвет, которым будут показаны все линии на данном слое.

Flash Color: Отображает цвет, которым будут показаны все флэши на данном слое.

ON: Управляет режимом отображения данного слоя.

Lock: Управляет режимом блокировки данного слоя. Если опция включена, то данный слой будет заблокирован, то есть его содержимое будет невозможно изменить, в частности, не будут работать функции рисования.

Щелчок правой кнопки мыши на заголовке любого из трех последних столбцов последовательно включает или выключает данную опцию для всех слоев в таблице.

Delete: Признак удаления слоя. Если эта опция включена, то при нажатии на кнопку **Apply** или **OK** данный слой будет удален из проекта.

При нажатии кнопки **Apply**, все изменения будут выполнены без закрытия диалогового окна. Это экономит время, при выполнении каких-либо действий с разными слоями. Порядок прорисовки слоев задается в диалоговом окне **View Layer Order** после нажатия кнопки **View Layer Order**.

Флаг **Draw Current Layer Last** включает режим (рис. 7.33), когда активный слой прорисовывается поверх всех остальных независимо от порядка их вывода (**Top to Bottom** или **Bottom to Top**). По умолчанию этот флаг всегда включен.

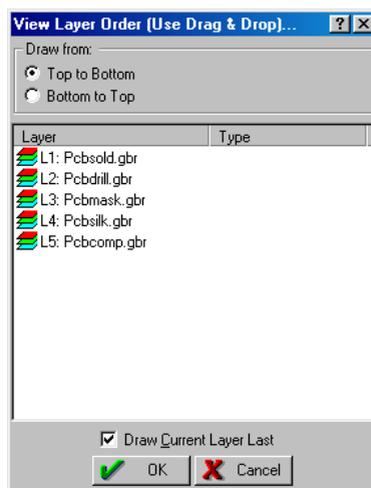


Рис. 7.33. Настройка порядка прорисовки слоев.

Данная опция может быть изменена в диалоговом окне **General Preferences**.

Режимы захвата и рисования

Панель инструментов **Coordinate** помимо списка слоев содержит кнопку **Drawing Modes** и выпадающие списки выбора единиц измерения (дюймы или миллиметры) и режимов захвата **Snap Mode** (рис. 7.34).

Если режим захвата отключен (**Off**), курсор будет просто указывать место, в котором будет выполняться функция рисования. При выборе любого другого режима, курсор меняет вид и в нем появляется крестик всякий раз, когда указатель мыши нацелен в нужное место. Здесь имеются следующие опции:

Off: Режим свободного перемещения курсора по окну проекта.

Grid: Перемещает курсор к ближайшей точке сетки.

Auto(Center+End): Перемещает курсор в конец ближайшей линии или центр окружности.

Center: Перемещает курсор в центр любого флэша.

CenterLine: Перемещает курсор по оси линии не зависимо от ее ширины.

Edge: Перемещает курсор по внешнему краю объекта.

End: Перемещает курсор в ближайший конец линии.

Intersection: Перемещает курсор в точку пересечения двух объектов (линий или дуг). При этом они могут иметь разную ширину и располагаться на разных слоях.

Mid: Перемещает курсор в середину любой линии или дуги.

Object-Center: Перемещает курсор в рассчитанный центр ("центр тяжести") дуги или полилинии. Эта точка для дуги не совпадает с центром соответствующей ей окружности, а характеризует реальные ее размеры.

Quadrant: Перемещает курсор на крайнюю левую, верхнюю, правую или нижнюю точку дуги или окружности.

Text Insert: Перемещает курсор на точку вставки любого существующего текста. В программе SAMtastic под текстом подразумевается только соответствующий объект. Если текст изображен последовательным набором линий или полилиний, этот метод не работает.

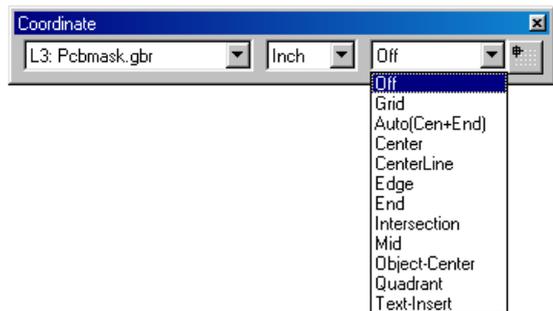


Рис. 7.34. Выбор режима захвата.

Нажатие на кнопку **Drawing Modes** вызывает одноименное диалоговое окно (рис. 7.35).

Поле **Grid** управляет режимом отображения сетки привязки. Шаг сетки задается в расположенном здесь выпадающем списке. В поле **Modes** опция **Ortho** включает ортогональный режим рисования линий. Опция **Blip** включает режим отображения маленького крестика в точке, где был выполнен щелчок левой кнопкой мыши без активной команды рисования. Все эти указательные точки являются временными и удаляются при обновлении экрана.

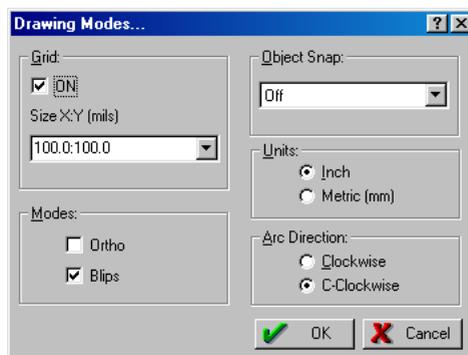


Рис. 7.35. Настройка режима рисования.

Здесь же имеется переключатель между дюймовой или метрической системами измерения, переключатель направления рисования дуг (по часовой стрелке или против нее) и описанный выше выпадающий список выбора режима захвата.

Таблица инструментов

Таблица инструментов содержит описание всех используемых сверел и фрез. Если вы при прорисовке на топологии задаете новые инструменты, эта таблица автоматически обновляет описания всех необходимых инструментов для сверления, фрезерования и трассировки. Иначе она будет оставаться пустой до тех пор, пока размеры инструментов не будут введены вручную. D-коды (начиная с первого доступного кода после 499) будут связаны с каждым инструментом.

Таблицу инструментов можно открыть на вкладке **Main View** панели управления **Project Workspace** (рис. 7.36).

Number (#): Номер инструмента (1-99).

Size: Диаметр сверла.

Comp: Компенсация реза.

Feed: Скорость механизма подачи.

Speed: Скорость шпинделя.

Z-Axis: Глубина смещения.

Plated: металлопокрытый или нет.

#	Size (Inch)	Comp. (Inch)	Feed (In/Min)	Speed (RPM)	Z-Axis (Inch)	Plated	Ref. Dcode	Comments
1	1.0000	1.0000				<input checked="" type="checkbox"/>	500	
2						<input checked="" type="checkbox"/>		
3						<input checked="" type="checkbox"/>		
4						<input checked="" type="checkbox"/>		
5						<input checked="" type="checkbox"/>		
6						<input checked="" type="checkbox"/>		
7						<input checked="" type="checkbox"/>		
8						<input checked="" type="checkbox"/>		
9						<input checked="" type="checkbox"/>		
10						<input checked="" type="checkbox"/>		
11						<input checked="" type="checkbox"/>		

Рис. 7.36. Таблица инструментов.

Ref.Dcode: Существующие D-коды используются для вывода пути трассы.

Comments: Комментарии.

Файлы инструментов могут быть созданы или импортированы, а в формате такого файла есть небольшая свобода действий. Ниже приведены ключевые компоненты файла таблицы настроек Mill/Route, который должен быть сохранен в ASCII формате.

[любой текст]

Ver=2

[любой текст]

Tool=1, [Size], [Comp.], [Feed], [Speed], [Z-Axis], [Tab], [Dcode], [Plated], [Comments]

Tool=2, и т.д.

Если какое-нибудь из этих значений неизвестно и не играет роли, в соответствующих полях вводятся нули (исключая **Comments**, которое может быть пустым). Для **Plated**: Не металлопокрытый – 0, Металлопокрытый – 1. **Ref.Dcode** не создает D-код, он просто предоставляет связи из своего выпадающего списка. Так, если основной D-код не связан с таблицей апертур, ячейка **Ref.Dcode** останется пустой, даже если его номер есть в файле *.MTS.

Режимы просмотра файлов

Иногда возникает необходимость в просмотре импортированных файлов. CAMtastic! 2000 Designers' Edition предлагает множество разнообразных инструментов для просмотра файлов. Из этих файлов можно сгенерировать несколько типов отчетов.

Рассмотрим инструменты просмотра.

Переключатель Toggle

Схему изображения можно увидеть, выполнив команду меню **View » Fill**. По умолчанию, переключатель **Toggle** является активным, это означает, что все точки и полигоны будут закрашены. Когда опция отключена, все точки и полигоны будут показываться как схема. Пользовательские апертуры не подчиняются этой опции.

Переключатель Negative

Инвертирование изображения можно выполнить с использованием команды меню **View » Negative**. Когда эта опция включена, изображение показывается инвертированным. Если в стандартном виде на обычном фоне

Имеется цветное изображение, то в инвертированном виде будет цветной фон с чистым изображением.

Переключатель Translucent

Эта функция показывает объекты прозрачными, позволяя увидеть объекты, частично или полностью перекрытые другими объектами. Для активирования функции выполните команду меню **View » Translucent**. Хотя эта функция полезна, она может доста-

вить немало хлопот. Составные объекты иногда показываются белым цветом, используемым только для выделения. Это может сбить с толку, поскольку можно подумать, что объекты выделены, в то время как на самом деле один объект перекрывает другой. К тому же скорость перерисовки изображения увеличивается, когда эта функция выключена.

В программе CAMtastic можно сравнить слои, выводя на экран только их различия. Включите слой, который Вы хотите вывести на экран, выполните команду меню **View » Difference**. Будут выведены только те элементы, которых нет ни на каком другом слое.

Функция **Highlight Dcode** показывает апертуры так, как если бы они были выделены, на самом деле это неверно, поскольку функция редактирования недоступна во время действия **Highlight Dcode**. Для редактирования этих элементов активизируйте функцию редактирования, нажмите кнопку **Filter** на панели инструментов **Select**, введите номер(а) D-кодов в верхнем поле и нажмите кнопку **OK**. Далее протащите окно выделения через требуемую область.

Все вхождения отдельных D-кодов можно подсвечить. Сначала проверьте, что текущий D-код является тем, который необходимо подсветить, затем выполните команду меню **View » Highlight Dcode**. Эта функция доступна только в режиме просмотра. Если включен режим редактирования или рисования, функция недоступна.

Переключатель Film Box

Для задания или изменения зармеров пленки выполните команду меню **View » Film Box**, которая открывает диалоговое окно, показанное на рис. 7.37. Если зармеры пленки были описаны ранее, выполните команду меню **View » Film Box**, которая выведет контур пленки на экран.

Хотя **Box Size** должен быть больше нуля и не более 65 дюймов, размеры **Lower Left Corner** могут быть положительными или отрицательными.

Доступен выпадающий список ранее описанных конфигураций пленки. Для описания новой конфигурации введите имя в поле рядом с **Box Name**, размеры и положение начала координат. Если вы хотите, чтобы это диалоговое окно появлялось при следующем запуске проекта CAMtastic, включите опцию **Display Film Box at StartUp**.

Окно проекта закрашивает все окно, но для черчения выделяет 65 x 65 дюймов, эта область называется **Extent Box**. Для отображения этой области выполните команду меню **View » Extent Box**. Если рабочая область не видна полностью в текущем масштабе, уменьшайте масштаб до тех пор, пока полностью не станет видна пунктирная линия. Следует помнить, что окно координат, расположенное с левой части строки состояния, показывает местоположение курсора как строку **Out of Limits!**, когда курсор перемещается за пределы рабочей области.

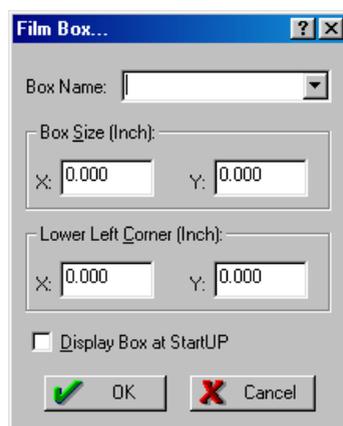


Рис. 7.37. Задание размеров пленки.

Начало координат обычно вне зоны видимости, для вывода на экран осей X и Y выполните команду меню **View » Origin**. После этого изображение переместится так, что на экране будет видна рабочая область размером 65 x 65 дюймов. Ее местоположение по умолчанию можно указать в **General Preferences**. Можно также изменить оригинальное начало координат, с помощью команды меню **Bonus » Set Origin**, после чего указать новое начало координат. Координаты, показываемые в окне координат, будут вычисляться относительно нового начала координат.

Нижний левый угол импортируемых файлов будет вставляться в начало координат, так что вы можете получить сообщение об ошибке, если начало координат расположено слишком близко к верхней правой границе рабочей области. Будет выдан запрос, следует ли подогнать начало координат так, чтобы документ поместился в рабочей области.

Окно проекта периодически перерисовывается, обычно после выполнения какого-либо действия. Экран можно автоматически перерисовать, выполнив команду меню **View » Redraw**. Все включенные слои перерисовываются в порядке, указанном в **Layer Table**.

Инструменты верификации

Программа САМtastic! 2000 Designers' Edition позволяет легко измерить расстояние между объектами или выполнить запрос параметров одного или нескольких объектов.

Измерения

Расстояние между точками можно измерить, выполнив команду меню **Inspect » Measure: Point to Point**. Далее укажите первую точку, а затем вторую.

Автоматически открывающаяся вкладка **Info View** панели управления **Project Workspace** покажет расстояние, угол, координаты каждой точки, X и Y смещение.

Для точного измерения координаты точек могут быть введены вручную в окне координат в строке состояния. Далее следует нажать клавишу Enter.

Выбор конкретных точек **Point 1** и **Point 2** при измерении зависит от выбранного режима привязки к сетке и шага.

Расстояние между объектами можно определить, выполнив команду меню **Inspect » Measure: Object to Object**. Результат появится на вкладке **Info View** панели управления **Project Workspace**.

Запросы

Пользователь может получить информацию об объектах топологии (флэшах и линиях) с помощью команды меню **Inspect » Query**, после чего курсор изменит форму на "указательный палец". В этом режиме можно выбрать одновременно только один объект. На вкладке **Info View** панели управления **Project Workspace** появится список, содержащий информацию об объекте: D-код, тип слоя, имя слоя и его номер, длина, имя цепи (если выполнялась экстракция списка соединений), форму, размер, тип и ближайшую вершину.

Аналогичным образом можно получить информацию о группе элементов топологии. Для этого используется команда меню **Inspect » Query Group**, после чего потребуется указать набор объектов щелчком левой кнопкой мыши или в окне выделения. Число выбранных объектов будет отображаться в поле **Total** в строке состояния. Щелчок правой кнопкой мыши прекращает процедуру выделения и запускает генерацию отчета, который отображается на вкладке **Info View** панели управления **Project Workspace** (рис. 7.38).

Если ранее выполнялась операция экстракции списка соединений, то можно получить дополнительную информацию о цепях, с помощью команды меню **Inspect » Query Net**. После указания любого объекта на топологии будет подсвечена все цепь, к которой принадлежит данный объектего, на панели управления **Project Workspace** будет показано имя указанной цепи, ее длина и информацию о слое (имя, номер и тип).

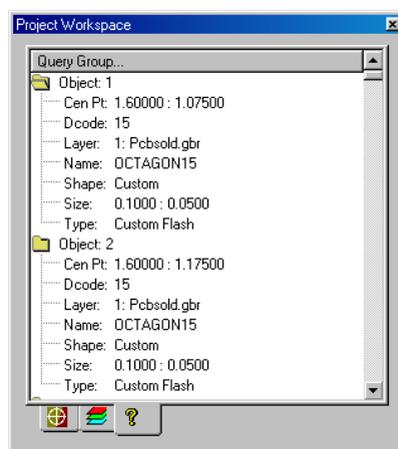


Рис. 7.38. Параметры группы объектов.

Экстракция списка соединений

Экстракция списка соединений позволяет получить информацию о цепях из связанных объектов, расположенных на разных слоях топологии. Генерация списка соединений выполняется с помощью команды меню **Inspect » Netlist Extraction**. Полученный список цепей отображается на вкладке **Netlist View** панели управления **Project Workspace** (рис. 7.39).

Перед выполнением этой операции убедитесь, что в проекте в окне **Layer Table** описаны следующие типы слоев: **Top**; **Bottom**; **Drill Top**; **Neg Plane**; **Pos Plane**.

Щелчок левой кнопкой мыши на любой папке с именем цепи приведет к ее подсветке и масштабированию в окне просмотра топологии. Для обновления списка соединений на панели управления **Project Workspace** используется команда меню **Inspect » View Nets**.

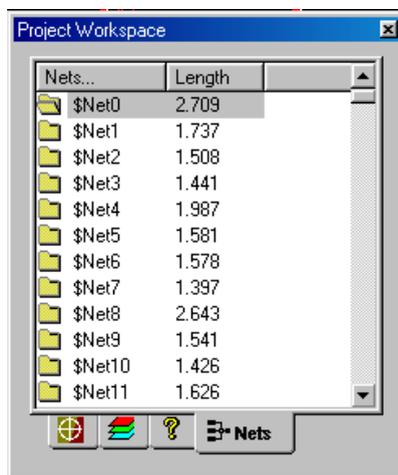


Рис. 7.39. Просмотр цепей.

Проверка правил проектирования и технологических норм (DRC/DFM)

В программе САМtastic имеются ряд инструментов, которые дают возможность проверить проект на предмет нарушения установленных правил проектирования. Проверка запускается командой меню **Inspect » DRC/DFM Check**. После указания набора

проверяемых объектов и щелчка правой кнопкой мыши в окне проекта откроется показанное на рис. 7.40 диалоговое окно.

Здесь задаются критерии проверки правил проектирования DRC, такие как минимально допустимые размеры контактных площадок и проводников, зазоры между различными объектами и соотношения размеров отверстий, контактных площадок и окон в трафарете маски.

Опции проверки технологических норм DFM приведены в разделе меню **DFM Checks** (рис. 7.41).

Power/Ground Short: замыкания на внутренних слоях питания и заземления.

Non-Functional Internal Pads: неиспользуемые контактные площадки на внутренних слоях.

Silkscreen Over SolderMask: наложение объектов, расположенных на слое шелкографии на объекты слоя маски пайки.

DrillVerification: Double Hits: наложение отверстий.

DrillVerification: Touching Holes: касающиеся отверстия.

Раздел меню **File** внутри этого диалогового окна позволяет открыть или сохранить настройки поля. Для запуска проверки нажмите кнопку **OK**. При отсутствии нарушений на экран будет выведено соответствующее сообщение. Если ошибки будут найдены, то на экран будет выведен список выявленных нарушений DFM. На панели управления **Project Workspace** станет доступной вкладка **Error View**, на которой все нарушения будут сгруппированы в папки по категориям.

В проект автоматически будет добавлен слой **DRCError**, на котором будут прорисованы маркеры ошибок. При выделении ошибки на панели управления **Project Workspace** соответствующий маркер будет подсвечен в поле просмотра проекта. Подробный отчет о выявленных ошибках DRC сохраняется в файле **DRCREPORT.TXT** в той же папке, где находится текущий проект.

В любое другое время список нарушений DRC можно просмотреть в панели управления **Project Workspace**, выполнив команду меню **Inspect » View Errors**. Следует помнить, что при этом повторная проверка проекта не производится. Ес-

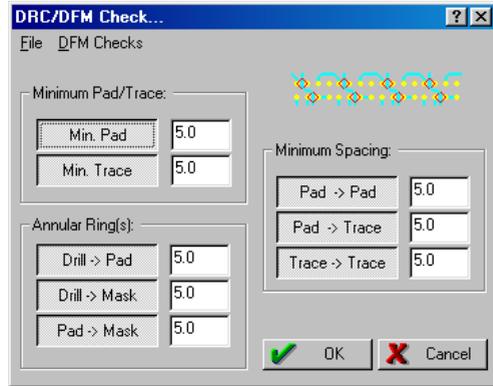


Рис. 7.40. Настройка критериев проверки DRC.



Рис. 7.41. Опции проверки DFM.

ли проверка DRC/DFM еще не выполнялась, будет выдано сообщение, что слой **DRCError** не найден.

Программа CAMtastic также имеет функцию проверки правильности прорисовки полигонов. Полигоны должны быть нарисованы замкнутым контуром, не перекрывать-ся и не иметь общих вершин. Все эти условия проверяются с помощью команды меню **Inspect » Invalid Polygon Check**. В случае обнаружения ошибок на экран будет выведено сообщение **Invalid Polygon Found**, а сам полигон будет подсвечен в окне просмотра проекта.

Подсчет площади металлизации

Для расчета площади металлизации используется команда меню **Inspect » Copper Area Calculation**. После выбора нужных объектов или всей платы и щелчка правой кнопкой мыши запустится процедура расчета площади металлизации для всех включенных в проекте слоев. Будет создан и выведен на экран файл отчета **COPPER.TXT**, в котором приведена суммарная площадь металлизации, а также расписаны площади на каждом включенном слое.

Генерация отчетов

Отчеты могут быть созданы в любое время работы над проектом. Список доступных отчетов представлен в папке **Reports** на главной вкладке панели управления **Project Workspace**.

Все отчеты сохраняются в файлы в ASCII формате в последней открытой программой CAMtastic папке. Если вы работаете с новым проектом, который еще ни разу не сохранялся, отчеты могут создаваться в другой папке. Для точного указания места хранения проекта и сгенерированных по нему отчетов необходимо использовать функцию **Save As**.

Некоторые отчеты получают имя, ассоциированное с активным проектом, другие отчеты имеют стандартные имена и хранятся в одной папке с проектом. Если в указанной папке уже существует отчет с таким именем, то новый отчет заменяет старый без предупреждения. Поэтому рекомендуется хранить все проекты в отдельных папках.

Апертуры

Для создания текстового файла отчета с таблицей апертур и вывода его на экран выполните двойной щелчок на опции **Apertures Report**. Файл отчета обычно включает в себя заголовок, настройки по умолчанию и список D-кодов с ассоциированными апертурами. Файл получит имя проекта и расширение **.LST** и станет доступным для

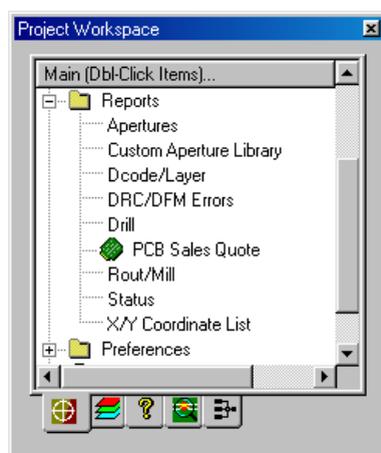


Рис. 7.42. Доступные отчеты.

импорта с помощью команды меню **File » Import » Aperture**. Ниже приведен пример такого файла.

```

CAM Products 2000 Aperture List
Serial Number: 0000-00-00000

Format: 3.4
Zero Suppression: Leading
Type: Absolute
UNITS: English (In.)

DCODE:          TYPE:          X : Y          BLOCK/CUSTOM:
-----
10              Round   0.0010:0.0010
11              Round   0.0010:0.0010
12              Round   0.0050:0.0050
13              Round   0.0070:0.0070
14              Custom           0.0000:0.0000  OCTAGON14
15              Custom           0.0000:0.0000  OCTAGON15
16              Round   0.0320:0.0320

```

Пользовательские апертуры

Для создания текстового файла, содержащего описание всех пользовательских апертур активного проекта, выполните двойной щелчок на опции **Custom Aperture Library**.

Файл получит имя проекта и расширение **.LIB** и станет доступным для импорта с помощью команды меню **File » Import » Custom Aperture Library File**. Пример такого файла приведен ниже.

```

%AMОCTAGON14*
4,1,8,0.055000,0.012400,0.037400,0.030000,-0.037400,0.030000,-
0.055000,0.012400,-0.055000,-0.012400,-0.037400,-0.030000,0.037400,-
0.030000,0.055000,-0.012400,0.055000,0.012400,0.0*
%
%AMОCTAGON15*
4,1,8,0.050000,0.010400,0.035400,0.025000,-0.035400,0.025000,-
0.050000,0.010400,-0.050000,-0.010400,-0.035400,-0.025000,0.035400,-
0.025000,0.050000,-0.010400,0.050000,0.010400,0.0*
%
%AMROUNDRECT30*
4,1,4,0.047000,0.047000,-0.047000,0.047000,-0.047000,-0.047000,0.047000,-
0.047000,0.047000,0.047000,0.0*
1,1,0.047000,0.047000,0.047000*
1,1,0.047000,-0.047000,0.047000*
1,1,0.047000,-0.047000,-0.047000*

```

```
1,1,0.047000,0.047000,-0.047000*
20,1,0.047000,0.047000,0.047000,-0.047000,0.047000,0.0*
20,1,0.047000,-0.047000,0.047000,-0.047000,-0.047000,0.0*
20,1,0.047000,-0.047000,-0.047000,0.047000,-0.047000,0.0*
20,1,0.047000,0.047000,-0.047000,0.047000,0.047000,0.0*
```

Апертуры на слоях

Для получения отчета об апертурах, используемых на конкретных слоях выполните двойной щелчок левой кнопкой мыши на опции **DCode/Layer**. Откроется окно **DCode/Layer Report** (рис. 7.43), в котором следует задать настроить генератор отчетов. Отчет может быть сгенерирован для всех без исключения слоев (**All Layers**), только отображаемых в данный момент слоев (**ON Layers**) и одного конкретного слоя. В верхней части окна приводится список используемых на слое апертур, содержащий номер D-кода, форму, размер, число флешей и линий, нарисованных с помощью данной апертуры. В нижней части окна приводится суммарная информация, систематизированная по слоям. После нажатия на кнопку Save будет сформирован и записан файл **REPORT.TXT**.

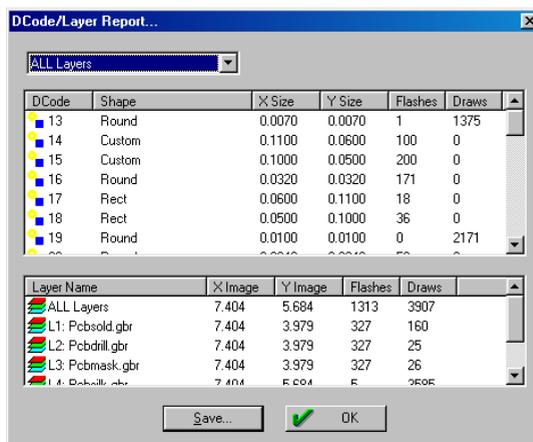


Рис. 7.43. Окно настройки отчета об апертурах, используемых на конкретных слоях.

Ошибки проверки DRC/DFM

Если для данного проекта уже выполнялась описанная выше проверка DRC/DFM, то файл отчета **DRCREPORT.TXT** уже сформирован. Двойной щелчок левой кнопкой мыши на опции **DRC/DFM Errors** просто откроет его.

Отверстия

Двойной щелчок левой кнопкой мыши на опции **Drill** запускает генерацию текстового файла **DRILL.RPT**, содержащего информацию о всех используемых на плате инструментах сверления, независимо от режима отображения соответствующих слоев. Здесь для каждого инструмента приводится диаметр и число проделанных им отверстий. Ниже приведен пример такого отчета.

```
CAM Products 2000 Drill Report
Serial Number:0000-00-00000
Units: English (Inch)
```

TOOL	SIZE	COUNT
=====	=====	=====
1	0.0540	42
2	0.0630	116
3	0.0625	5
4	0.1400	6
5	0.0500	10
6	0.0750	24
7	0.1500	2
8	0.4000	2
9	0.2000	2

Отчет для оценки стоимости изготовления платы

Программа САМtastic дает возможность сформировать специальный отчет для отправки на производство для оценки стоимости изготовления печатной платы, включающий размеры, минимальные ширины проводников и зазоров, число контактных площадок и отверстий.

Двойной щелчок левой кнопкой мыши на опции **PCB Sale Quote** открывает окно, показанное на рис. 7.44. Для построения отчета должен быть определен хотя бы один трассировочный слой.

Кнопка **Select** в поле **PCB Border** предназначена для определения контура платы.

Кнопка **Select** в поле **Minimum Trace/Gap** предназначена для указания группы объектов для определения минимальной ширины проводников и зазоров.

Кнопка **Option** открывает диалоговое окно **PCB Report Option** (рис. 7.45). Здесь имеются более 30 опций, включающих в отчет информацию, которая может потребоваться для оценки стоимости изготовления. После выбора необходимых опций необходимо выбрать единицы измерения в списке **Report Units** и размер листа материала в поле **Panel Size**. Нажатие кнопки **OK** вернет вас в предыдущее диалоговое окно.

После задания всех необходимых параметров следует нажать кнопку **OK** в диалоговом окне **PCB Sales Quote**. Будет сформирован и открыт текстовый файл **PCBREPORT.TXT**. Пример такого отчета приведен ниже.

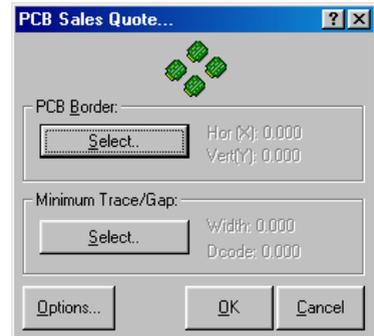


Рис. 7.44. Окно PCB Sales Quote.

```
CAMtastic! PCB Report File: Demopcb
Date:                10/27/2002
Time:                20:16:3
Units:               Inch

Dimensions:
=====
Horizontal (X):      7.060
Vertical (Y):        3.060
Total PCB Area (Sq.In): 21.604
Perimeter (Length): 20.240

Panel Size/Count:
=====
PCB Panel Size (X:Y) (in.): 18:24
Total Count per Panel: 15
X (Horizontal) Count: 3
Y (Vertical) Count: 5
```

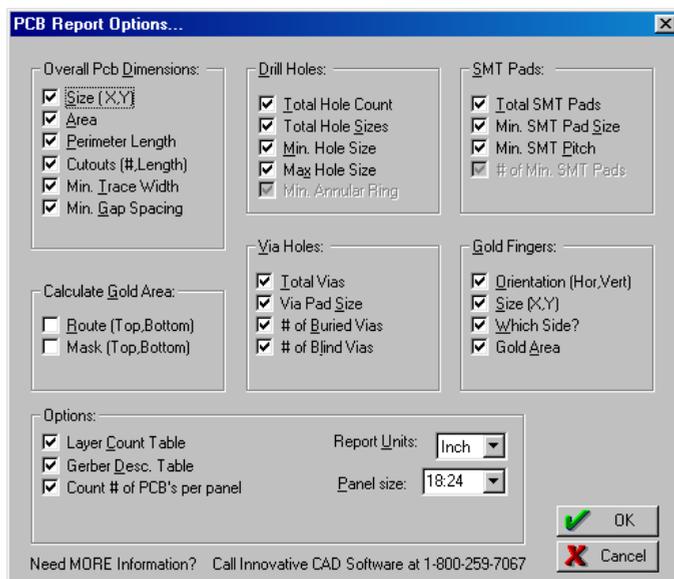


Рис. 7.45. Окно PCB Sales Quote.

Фрезеровка

Двойной щелчок левой кнопкой мыши на опции **Rout/Mill** запускает генерацию текстового файла **ROUT.RPT**, содержащего описания всех фрез, число и длину трасс фрезерования.

```

CAM Products 2000 Rout-Mill Report
Serial Number:0000-00-00000
Units: English (Inch)

TOOL          SIZE    COUNT  DIST
=====
1             0.0100  1      5.5770

```

Системная информация

Запуск отчета **Status**, в отличие от других отчетов, не создает текстового файла. Двойной щелчок левой кнопкой мыши на опции **Status** открывает одноименное диалоговое окно (рис. 7.46), отображающее информацию о проекте, а также системную информацию.

Список X/Y координат

После двойного щелчка левой кнопкой мыши на опции **X/Y Coordinate List** пользователь должен указать область топологии, для которой должен быть сформирован отчет. После завершения выделения будет автоматически создан текстовый файл **XYLIST.RPT**, содержащий координаты всех выделенных объектов. Пример такого отчета приведен ниже.

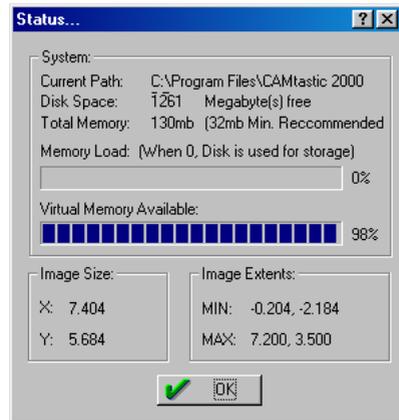


Рис. 7.46. Окно отчета Status.

```

CAM Products 2000 X/Y Report
Serial Number:0000-00-00000
Units: English (Inch)

Total Objects Selected: 141

X1,Y1,X2,Y2,Start Angle,End Angle,Radius,Bulge
=====
-7.000000,-5.825000
-7.000000,-5.625000
-7.000000,-5.725000
-7.200000,-5.725000
-7.200000,-5.625000

```

```
-7.200000,-5.825000
-6.725000,-5.250000
-7.406000,-4.744000
-6.600000,-4.750000
Trace: Vertex Count=1
-7.200000,-5.625000,-7.000000,-5.625000
Trace: Vertex Count=1
-7.000000,-5.625000,-7.200000,-5.625000
```

Инструменты редактирования

Перед рассмотрением инструментов редактирования следует три важных момента:

- Для отмены последних выполненных изменений может быть использована многоуровневая система отката с помощью команды **Edit » Undo**. Аналогичным образом работает команда **Edit » Redo**, отменяющая действие команды **Undo**.
- По мере обработки CAD файлы имеют тенденцию увеличиваться в размере. Определенные возможности, такие, как масштабирование и массивы данных, реализует функция **Step & Repeat**, которая использует циклический код вместо перезаписывания последовательности в сохраненном файле. Если этот код используется в проекте, любой Gerber или NC Drill/Mill файл также будет использоваться его, что улучшит производительность фотопостроителей, сверлильных и фрезерных станков. Для включения функции **Step & Repeat** в проект используются команды из подменю **Bonus » Step & Repeat**.
- Некоторые процедуры требуют наличия замкнутой линии, что означает больше чем просто сходящиеся или пересекающиеся линии. В программе CAMtastic автоматически замкнутыми могут быть только полилинии и контуры полигонов.

Рисование

Любые инструменты рисования используют текущий D-код, отображаемый на панели **Aperture**.

Флэши

Одиночное размещение любой апертуры называется флэшем. Флэш можно поставить в любом месте рабочей области среды проектирования с помощью команды меню **Draw » Flash**. Далее следует щелкнуть левой кнопкой мыши в требуемой точке топологии или, для более точного размещения, ввести координаты в поле в строке состояния и нажать **Enter**.

Линии

Для прорисовки простых линий используется команда меню **Draw » Line**. Для рисования замкнутых линий следует использовать команду **Draw » Polyline**. Ломаная линия, нарисованная с помощью простых линий, в отличие от полилиний, допускает изменение D-кода и слоя для отдельных ее сегментов.

После выбора нужной команды следует указать начальную точку. Следующая точка будет очередной вершиной и так далее, до завершения линии. Щелчок правой кнопкой мыши или нажатие клавиши **Esc** заканчивает рисование линии.

Рисование полилиний выполняется аналогично, но при нажатии правой кнопки мыши появляется контекстное меню, предлагающее следующие опции рисования:

End: Завершает линию без ее закрытия.

Close: Рисует конечную линию из последней вершины в текущую вершину, замыкает линию и завершает рисование линии.

Line: Переключает режим рисования из режима прямой линии в режим дуги.

Arc: Переключает режим рисования из режима дуги в режим прямой линии.

Clockwise: Рисует дугу из последней вершины по часовой стрелке.

C-Clockwise: Рисует дугу из последней вершины против часовой стрелки.

Undo: Отменяет ввод предыдущей вершины.

Snap: Включает или выключает режим привязки к узлам сетки во время рисования полилинии.

Chancel: Отменяет все поставленные вершины во время рисования полилинии. После обновления экрана линия исчезнет.

Любая замкнутая ломаная линия может быть преобразована в полилинию с помощью команды **Join** » **Single**. Наоборот, любая полилиния может быть превращена в ломаную с помощью команды **Explode** » **Polylines**.

Если требуется нарисовать серию линий, следует выполнить щелчок правой кнопкой мыши в рабочем окне среды проектирования и в появившемся контекстном меню выбрать команду **Repeat** — будет запущена последняя команда рисования или редактирования.

Прямоугольники

Для рисования прямоугольников используется команда **Draw** » **Rectangle**, после чего необходимо указать два угла. Прямоугольник будет нарисован текущим D-кодом на активном слое.

Дуги

Дугу можно рисовать, как часть полилинии или отдельно. В программе CAMtastic имеется несколько различных методов рисования дуг. В подменю **Draw** » **Arc** доступны следующие команды:

Команда **Center, Start, Sweep Angle** требует указания центра и начальной точки дуги, далее просит ввести угол развертки (положительное значение для развертки против часовой стрелки, отрицательное для развертки по часовой стрелке).

Команда **Center, Start, End** требует указания центра, начальной и конечной точек дуги.

Команда **2-Points** рисует дугу 90° между двумя точками.

Команда **3-Points(Start, End, Tangent)** требует указания начальной и конечной точек, а далее показывает потенциально возможную дугу между ними при перемещении курсора мыши. При достижении необходимой формы щелкните левой кнопкой мыши или нажмите **Enter**. Данная функция также доступна из панели инструментов **Draw**.

Команда **Tangent, Tangent, Radius** после выбора двух не параллельных линий требует ввести радиус. Дуга будет нарисована между линиями и будет выпуклой в сторону их пересечения. Если дуга рисуется для сегментов полилиний, то могут быть указаны только сосерние сегменты, в противном случае полилинию следует преобразовать в ломаную.

Для команд **Center, Start, End** и **2-Points** направление дуги задается в диалоговом окне **Drawing Modes**.

Следует также помнить, что критические точки любой дуги могут быть точно определены путем ввода их координат в поле координат в строке состояния.

Окружности

Как и в случае с рисованием дуг, имеется несколько различных методов рисования окружностей. Все методы используют текущие D-код и слой.

В подменю **Draw » Circle** доступны следующие команды:

Команда **Center, Tangent Point** требует указать центр окружности, затем точку периметра. Эта функция также доступна в панели инструментов **Draw**.

Команда **Center, Radius** требует указать центр окружности, а затем ввести радиус.

Команда **Tangent, Tangent, Radius** после выбора двух не параллельных линий требует ввести радиус. Если окружность рисуется для сегментов полилиний, то могут быть указаны только сосерние сегменты, в противном случае полилинию следует преобразовать в ломаную.

Команда **2-Points** требует указать две любые точки периметра, определяющие диаметр окружности.

Команда **3-Points** требует указать три любые точки периметра. Эти точки не должны лежать на одной прямой.

Эллипсы

Для рисования эллипса следует использовать команду меню **Draw » Ellipse**, затем ввести два противоположных угла, создающих прямоугольник, который ограничивает эллипс. После того, как будет указан первый угол, на экране появится эллипс, который можно изменять движением курсора мыши.

Полигоны

Полигоном называется закрашенная область, ограниченная замкнутым контуром, имеющим как минимум три вершины. Для прорисовки полигонов используется команда меню **Draw » Polygon**, после чего указывается начальная точка, затем все последующие точки. Щелчок правой кнопки замыкает контур и завершает рисование полигона.

Текстовые надписи

Для выполнения текстовых надписей в программе SAMtastic можно использовать векторные или TrueType шрифты. После выполнения команды меню **Draw** » **Text** следует указать точку вставки. Откроется диалоговое окно **Draw Text**, как показано на рис 7.47. Здесь в поле **Font** по умолчанию выбрана опция **Use Standard**. В этом случае при нажатии на кнопку **Select** можно выбрать нужный файл шрифтов системы AutoCAD с расширением **.SHX**. Если выбрана опция **Use TrueType**, при нажатии на кнопку **Select** появится стандартное диалоговое окно выбора шрифта. Заметим, что программа SAMtastic работает только с символами с первой кодовой страницы шрифта.

Здесь также задаются:

- **Mirror**: Зеркальное отображение текста.
- **Italic**: Текст будет написан курсивом.
- Выпадающий список задает стиль выравнивания текста: **Top-Left**, **Top-Center** и т.д.
- **Height**: Задает высоту текста.
- **Rotate**: Задает угол поворота текста.
- **X Scale**: Задает масштабный коэффициент по оси X.
- Поле **Message** служит для ввода текста.
- Кнопка **Import Text** позволяет вставить текст из любого текстового документа.

При выборе опции **Use TrueType**, станут доступны три дополнительных поля:

- **Bold**: Текст размера будет написан полужирным начернением.
- **Fill Font**: Закрашивает каждый символ растровой заливкой.
- **Quality**: Управляет качеством шрифта.

Размеры

Программа SAMtastic имеет функцию простановки размеров. Однако, сначала следует проверить настройки шрифтов и точность. Эти настройки делаются в диалоговом окне **Dimension Setting** (рис. 7.48), вызываемом командой меню **Draw** » **Settings**.

Здесь задаются:

- **Font**: Позволяет выбрать файл шрифта.
- **Mirror**: Зеркальное отображение текста размера.

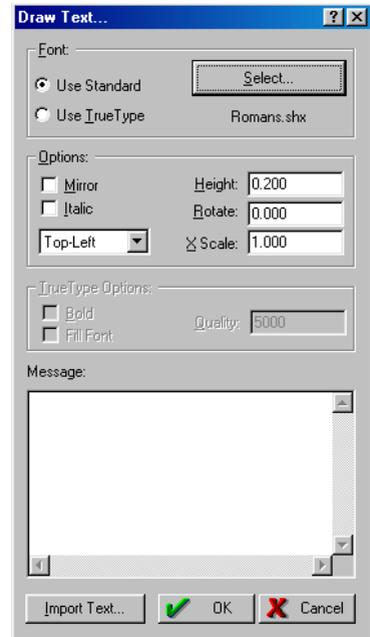


Рис. 7.47. Окно ввода текстовых надписей.

- **Bold:** Текст размера будет написан полужирным начертанием.
- **Italic:** Текст размера будет написан курсивом.
- **Upside Down:** Текст размера будет повернут на 180°.
- **Fill Font:** Закрашивает каждый символ текста размера растровой заливкой.
- **Height:** Устанавливает высоту текста размера.
- **X Scale:** Задаёт масштабный коэффициент по оси X.
- **Quality:** Управляет качеством шрифта текста размера.
- **Accuracy:** Задаёт число знаков после запятой в тексте размера.

Когда все настройки выполнены, простановка размеров производится с помощью команд из подменю **Draw » Dimension:**

Ordinate: Выводит размер с X и Y координатами. Горизонтальный размер будет показывать значение Y, вертикальный размер будет показывать значение X.

Horizontal: Создает размер, показывающий горизонтальное расстояние между двумя точками.

Vertical: Создает размер, показывающий вертикальное расстояние между двумя точками.

Radius: Создает размер, показывающий радиус дуги или окружности.

Редактирование

Вставка массивов объектов

Размещение массивов объектов выполняется с помощью команды **Edit » Array**. После выделения копируемых объектов откроется диалоговое окно **Array**, показанное на рис. 7.49.

Массив может быть построен двумя методами: **Distance** и **Gap**. Если выбрана опция **Distance**, будут доступны поля **Number of Columns**, **Number of Rows**, X и Y **Distance**. То есть объекты будут вставляться в ячейки таблицы, содержащей заданное число строк и столбцов определенного размера.

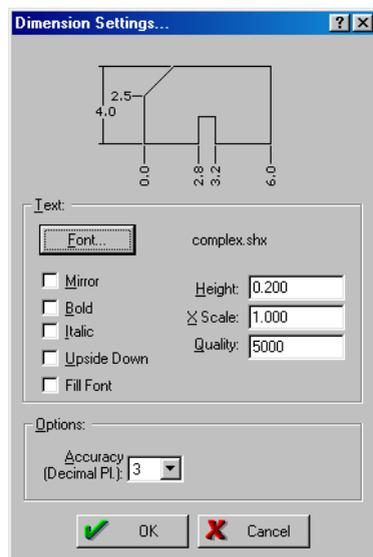


Рис. 7.48. Настройка шрифта для прорисовки размеров.

Если выбрана опция **Gap**, активными будут поля **Image Size** и **Gap**. Значение **Gap** задает расстояние между копиями объектов, а **Image Size** получается из размеров выделенной области по горизонтали и вертикали. Для получения дополнительного свободного места вокруг выбранного изображения значение **Image Size** можно увеличить вручную.

Здесь также имеется возможность включить опцию **Use Step & Repeat Codes**, что позволяет значительно уменьшить размер файла проекта и увеличить производительность программы.

Для предварительного просмотра результата размножения используют кнопку **Preview**.

Разбиение линии

Любая прямая линия может быть поделена на два отдельных сегмента с помощью команды меню **Edit » Break**.

Прорисовка фасок

Прорисовка фасок для двух соседних линий может быть выполнена с помощью команды меню **Edit » Chamfer**. В появившемся диалоговом окне необходимо через двоеточие ввести размеры фаски (расстояние по X и Y от точки пересечения, на котором будет нарисована новая соединяющая линия). Затем следует указать первую и вторую линии. Размер фаски не может превышать размеры выделенных линий.

Копирование

В программе SAMtastic объекты можно копировать и вставлять в проект неограниченное число раз. Для однократного копирования объекта в буфер обмена используется команда меню **Edit » Copy » Single**. Для многократного копирования используют команду **Edit » Copy » Multiple**. Копируемые объекты могут размещаться на разных слоях, при вставке каждый из них будет вставлен на свой слой.

Если требуется вставить объекты с одного слоя на другой слой, следует использовать команду **Edit » Copy to Layers**. После завершения выделения пользователю будет предложено выбрать слой для вставки. Если будет выбрана опция **All**, то объект будет вставлен на все без исключения слои.

Удаление

Удаление объектов выполняется с помощью команды **Edit » Erase**. Выбор нужного набора объектов можно выполнять с помощью инструментов, представленных на панели **Select**. Для удаления одного сегмента полилинии ее сначала необходимо преобразовать в ломаную линию с помощью инструментов **Break** или **Explode Polyline**.

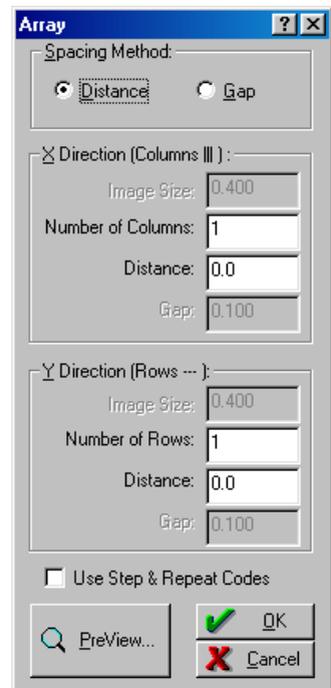


Рис. 7.49. Размещение массивов объектов.

Разбиение сложных объектов

Полилинии, текст и пользовательские апертуры в программе CAMtastic рассматриваются как цельные объекты. В отдельных случаях требуется разгруппировать эти объекты на отдельные части. Такое разбиение можно выполнить с помощью команд **Polylines**, **Text** и **Custom Apertures**, расположенных в подменю **Edit » Explode** вы увидите подменю из элементов. При разбивке текста он преобразуется в полилинии. Полилинии и пользовательские апертуры преобразуются в надоры отдельных линий и флэшшей.

Разбиению могут быть подвержены объекты, нарисованные с применением кода **Step & Repeat**, что равноценно удалению циклического кода. Подробно жта операция будет описана несколько позднее.

Автоматическое удлинение линий

Любая линия может быть продолжена до пересечения с другой линией с помощью команды **Edit » Extend**. Далее следует указать граничную и удлиняемую линию. Если пересечение двух указанных линий будет невозможно, система выдаст сообщение об ошибке.

Полилинии могут использоваться в качестве граничных линий, но не могут быть удлинены без предварительного разбиения с помощью команды **Explode**. Удлинение дуг не допускается.

Сопряжение линий

Сопряжение двух линий выполняется с помощью команды **Edit » Fillet**. После указания двух линий пользователю будет предложено ввести радиус закругления. Радиус следует указывать осторожно, так как при больших значениях дуга может не пересечься с исходными линиями.

Объединение линий

Отдельные линии и полилинии могут быть объединены с помощью команд из подменю **Edit » Join**. Команда **Single** преобразует в полилинию все сегменты ломаной после щелчка левой кнопкой мыши на любом из них. Команда **Group** предлагает более широкий набор возможностей объединения, задаваемых в диалоговом окне **Group: Join & Fix Polylines** (рис. 7.50).

Tolerance: Задает допуск на расстояние между концами отдельных линий и дуг, которые должны быть объединены.

Fuzzy Join Method: Находит ближайшие конечные точки линии с учетом значения **Tolerance**.

Intersection Join Method: Автоматически вычисляет точку пересечения двух соседних объектов, что позволяет создавать идеальные замкнутые объекты.

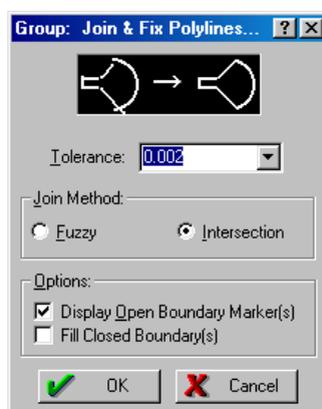


Рис. 7.50. Настройка объединения объектов.

Display Open Boundary Markers: Включает добавление маркеров на незамкнутые концы полилиний. Маркеры создаются на новом слое **Polyline PRO Open Markers**, который является временным и в проекте не сохраняется.

Fill Closed Boundary(s): Предписывает выполнить заливку замкнутого контура из полилинии.

Объединение будет выполнено после нажатия кнопки **OK**. Если некоторые границы остались незамкнутыми, рекомендуется повторить операцию с увеличенным допуском.

Зеркальное отображение

Зеркальное отображение объектов выполняется с помощью команды меню **Edit » Mirror**. После завершения выделения нажатием на правой кнопки мыши следует задать горизонтальную или вертикальную ось, относительно которой будет выполняться отображение. Далее последует запрос, надо ли удалить оригинал.

Изменение объектов

Программа САМtastic позволяет изменять отдельные атрибуты объектов, например, D-код и слой с помощью команды **Edit » Modify/Change**. После выбора объектов и нажатия правой кнопки мыши на экране появится диалоговое окно **Modify/Change** (рис. 7.51).

Опция **Use Dcode** включает изменение D-кода для всех выделенных объектов на другой из числа имеющихся. В противном случае потребуются выбрать форму апертуры и ее размеры.

Если требуется переместить выбранные объекты на новый или существующий слой, его необходимо указать в выпадающем списке **New Layer**. Для создания нового слоя необходимо выбрать в этом списке опцию **New Layer**. Программа автоматически создаст новый слой, присвоит ему имя по умолчанию, которое можно будет позже изменить на панели управления или в таблице слоев.

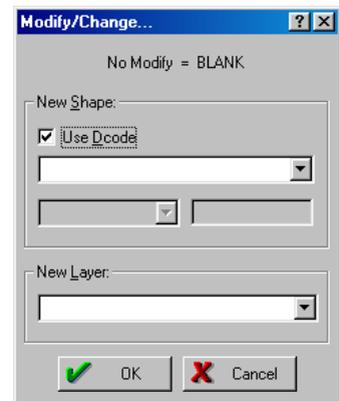


Рис. 7.51. Окно редактирования атрибутов объектов.

Перемещение объектов

Перемещение объектов на данном слое выполняется с помощью команды меню **Edit » Move**. После выделения нужного набора объектов они будут охвачены белым прямоугольником. Захватите этот прямоугольник и переместите в нужное место, удерживая левую кнопку мыши.

Построение эквидистантных объектов

Построение эквидистантных объектов относительно одиночных линий и полилиний выполняется с помощью команды меню **Edit » Offset**. Программа запросит ввести расстояние, на котором должен быть расположен эквидистантный объект, и указать направление — сторону исходной линии. Если полилиния замкнута, то эта функция

приведет к прорисовке внешнего или внутреннего контура, в зависимости от того, какая сторона была указана.

Поворот объектов

Поворот объектов или групп объектов выполняется командой меню **Edit » Rotate**. После задания группы объектов программа запросит указать центр вращения и ввести угол поворота. Для поворота против часовой стрелки надо ввести положительное число, для поворота по часовой стрелке — отрицательное. Для более точного указания центра вращения необходимо использовать поля ввода координат, расположенные в строке состояния.

Масштабирование

Масштабирование отдельных объектов и групп объектов выполняется с помощью команды меню **Edit » Scale**. После выделения нужных объектов программа запросит указать базовую точку, для чего можно использовать мышь или поля ввода координат в строке состояния. Далее появится диалоговое окно, в котором следует ввести коэффициент масштабирования в процентах. Программа также запросит подтвердить масштабирование апертур.

Редактирование текста

Размещенный в проекте текст можно изменить с помощью команды **Edit » Text**. После указания текстовой надписи появится описанное ранее диалоговое окно **Draw Text**. Внесите соответствующие изменения и нажмите кнопку **OK**.

Заметим, что если текст был преобразован в полилинии, то модифицировать его с помощью данной функции невозможно.

Добавление и удаление вершин

Программа САМtastic позволяет добавлять и удалять вершины у ломаных кривых и полилиний с помощью команд меню **Edit » Vertex » Add** и **Edit » Vertex » Delete** соответственно. Перемещение вершины полилинии выполняется командой меню **Edit » Vertex » Move**. Не допускается перемещение вершин, являющихся концами дуг.

Совмещение слоев

Если изображения на слоях имеют на разных слоях одинаковые рамки и объекты Target, то совмещение слоев может быть выполнено с помощью команды **Bonus » Align Layers » Border (Auto-Detect)**. Контур платы при этом должен быть прямоугольным и замкнутым. После указания контура все все слои, содержащие аналогичный контур будут автоматически совмещены.

Совмещение отдельных слоев выполняется командой **Bonus » Align Layers » Selective**. Далее следует выбрать опорный объект на первом слое, затем опорный объект на втором слое. Для правильного совмещения слоев необходимо выбирать одинаковые объекты, например левый нижний угол контура платы.

Создание перечня элементов

Программа САМtastic позволяет присваивать группировать наборы объектов на слоях в компоненты, присваивать им позиционные обозначения и генерировать перечень элементов для последующей сборки. Группировка объектов в компонент выполняется

няется командой **Tools » Assembly Utilities » Create & Group Parts**. Далее следует выделить требуемые объекты, после чего на экране появится диалоговое окно **Assign/Group Parts** (рис. 7.52). Выделяемые объекты должны находиться только на одном слое (например, шелкографии), для чего рекомендуется выключить все другие слои. Кроме того, выделение не может включать ранее сгруппированный компонент.

Description: Задаёт название создаваемого компонента.

Rotation (Deg.): Задаёт угол поворота компонента в градусах.

Board Side: Задаёт сторону платы, на которой размещается компонент (**Top**, **Thru** или **Bottom**).

После нажатия кнопки ОК программа автоматически найдет все объекты аналогичной формы и преобразует их в компоненты. На экране при этом появится закрашенный перечеркнутый прямоугольник. Система производит поиск в пределах слоя и различает только точные совпадения (форма, размер угол поворота).

Сформированным компонентам помимо названия может быть присвоено позиционное обозначение, для чего необходимо выполнить команду **Tools » Assembly Utilities » Assign Ref Designators**.

Список всех сформированных компонентов можно посмотреть выполнив двойной щелчок левой кнопкой мыши на опции **Assembly Parts Table** в панели управления. На экране появится диалоговое окно **Assembly Parts Table** (рис. 7.53), в которой будет приведена следующая информация: позиционное обозначение, координаты, сторона платы, угол поворота и имя компонента.

Кнопка **Export List** даёт возможность сохранить данный перечень элементов в текстовый файл с расширением **.PCF**.

Поля в таблице могут быть изменены, но это не окажет никакого влияния на объекты топологии, а после изменения все изменения будут утеряны. Тем не менее, измененные значения могут быть экспортированы во внешний файл.



Рис. 7.52. Группировка объектов в компонент.

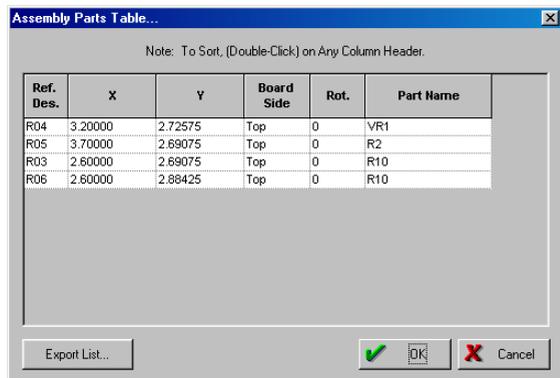


Рис. 7.53. Формирование перечня элементов.

Слияние слоев

Создать новый слой, содержащий копии всех объектов с набора существующих слоев можно с помощью команды **Bonus » Merge Layers**. Появится приведенное на рис. 7.54 диалоговое окно, в котором необходимо ввести название нового слоя и выбрать из списка исходные слои. Кнопка **All** служит для выполнения слияния всех без исключения слоев.

Композитные слои

Другая функция программы CAMtastic позволяет создавать так называемые композитные слои. Эта функция позволяет не просто объединять объекты, но добавлять и вычитать объекты один из другого, например, для создания вырезов внутри областей заливки. Кроме того, композитный слой может быть разложен обратно на исходные слои, что невозможно при простом слиянии слоев.

Для создания композитного слоя из произвольного набора объектов необходимо выполнить команду меню **Bonus » Composite Layers » Auto Create Composite**. Далее следует выбрать нужные объекты. Нажатие правой кнопкой мыши завершит выделение и вызовет диалоговое окно, приведенное на рис. 7.55. Здесь задается полярность композитного слоя: **Positive** (белый фон, черное изображение) или **Negative** (черный фон, белое изображение). Новый слой получит имя по умолчанию, которое позднее можно будет без труда изменить.

Для построения композитного слоя из набора существующих слоев используется команда меню **Bonus » Composite Layers » Build Composite**. Появится диалоговое окно, показанное на рис. 7.56, где задаются:

New Layer: Задаёт новое имя для композитного слоя.

Positive/Negative: Переключатель полярности композитного изображения.

Select Layer(s): Здесь выбираются исходные слои из списка существующих.

Polarity: Переключатель полярности **Dark/Clear** объектов данного слоя.

Spread (per side): Задаёт коэффициент масштабирования апертур на новом слое.

Все приведенные в этом окне слои обрабатываются именно в том порядке, в котором заданы. Также не рекомендуется использовать для создания композитных слоев ранее созданные композитные слои.

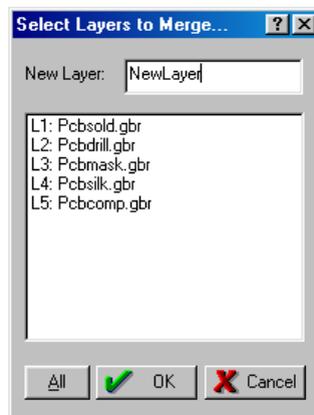


Рис. 7.54. Простое слияние слоев.

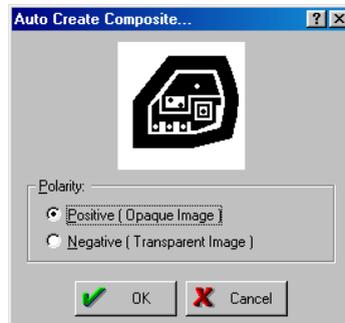


Рис. 7.55. Окно Auto Create Composite.

Созданный композитный слой может быть изменен с помощью команды **Bonus » Composite Layers » Modify Composite**. Далее следует указать любой объект на композитном слое, после чего заново откроется диалоговое окно **Composite Merge**. Однако слои здесь будут иметь новые имена **C1, C2** и т.д. Любой из этих слоев может быть удален из набора выбором пустой строки из выпадающего списка.

Разбиение композитного слоя на внутренние слои выполняется командой **Bonus » Composite Layers » Explode Composite**. Далее следует указать любой объект на композитном слое. Все слои **C1, C2** и т.д. будут преобразованы в новые слои с производными именами, например, **Name-1, Name-2** и т.д. в зависимости от имени композитного слоя. Сам композитный слой удаляется, причем эта операция необратима.

Для временного разбиения композитного слоя используется команда **Bonus » Composite Layers » Open Composite for Editing**. Эта команда работает аналогично **Explode Composite**, но после завершения редактирования композитный слой можно снова собрать командой меню **Bonus » Composite Layers » Close Composite**.

Заливка медью

Любая замкнутая фигура может быть залита медью, то есть внутри нее может быть создана область сплошной металлизации. Эта операция выполняется с помощью команды меню **Tools » Copper Pour**. Следует помнить, что в программе САМtastic понятие замкнутости контура подразумевает полное совпадение его вершин. Единственными всегда замкнутыми фигурами являются окружности и прямоугольники. Полилинии могут быть замкнутыми и незамкнутыми, в зависимости от того, как они были нарисованы. Если контур слагается из несгруппированных линий, перед заливкой медью их необходимо объединить в одну замкнутую полилинию.

После щелчка левой кнопкой мыши на замкнутой фигуре, появится диалоговое окно, приведенное на рис. 7.57.

В поле **Clearance** задается зазор между областью заливки и контуром. Отрицательные значения не допускаются.

В поле **Pattern** показывается последний использованный шаблон рисунка заливки, который можно изменить, нажав кнопку **Edit Pattern**. Появится диалоговое окно, показанное на рисунке 7.58.

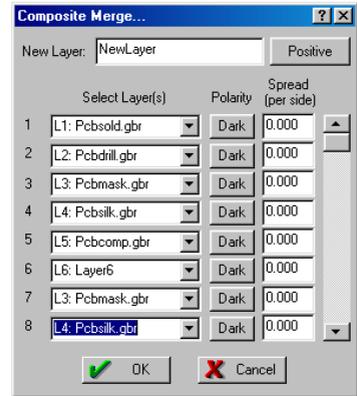


Рис. 7.56. Композитное слияние слоев.

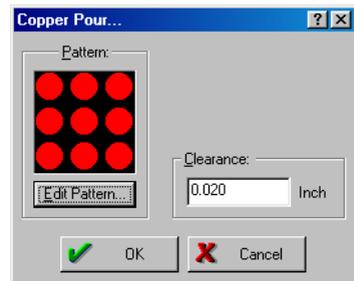


Рис. 7.57. Окно Copper Pour.

Если здесь установлен растровый способ заливки **Polygon (Raster)**, то все остальные настройки будут отключены. Заливка будет выполнена сплошной с учетом значения **Clearance**. При этом в список апертур будет добавлен новый D-код типа **Poex**.

Если установлен векторный способ заливки **Vector**, то рисунок может быть выполнен сплошным **Solid** или с узором **Shape/Dcode** с помощью имеющихся в проекте D-кодов или новой апертуры. При прорисовке узора в поле **X/Y Distance** задается шаг между отдельными его элементами.

Следует помнить, что использование растровой заливки значительно сокращает объем проекта.

Преобразование объектов

Программа CAMtastic имеет функции преобразования объектов топологии. Любой повторяющийся в проекте объект и группа объектов, могут быть заменены одной апертурой различными способами. Например, с помощью команды **Bonus » Draw to Flash » Automatic** можно указать объекты, которые нужно преобразовать во флэши. Выберите объекты, которые необходимо преобразовать. Помните, что при этом можно пользоваться кнопкой **Filter** на панели инструментов **Select**. После нажатия правой кнопки мыши появится приведенное диалоговое окно на рис. 7.59.

Если опция **Use Dcode** выключена, то для выделенных объектов будет создана новая апертура заданной формы. В противном случае, все выделенные объекты независимо от пита и размера будут заменены на указанный в выпадающем списке D- код.

Новые флэши могут быть созданы на новом, этом же или любом из существующих слоев. Кроме того, можно удалить старые объекты включив опцию **Delete Old Objects**. Также здесь задается допуск **Flash Tolerance** на поиск повторяющихся в выделении объектов и величину расширения апертуры **Over/Under Size**.

Другая функция преобразования объектов выполняемая с помощью команды меню **Bonus » Draw To Flash » Selective** предлагает принципиально другой подход. Сначала потребуется выделить одиночный объект, после чего на экране появится окно, аналогичное показанному на рис. 7.59. Но здесь рядом с выпадающим списком выбора формы апертуры име-

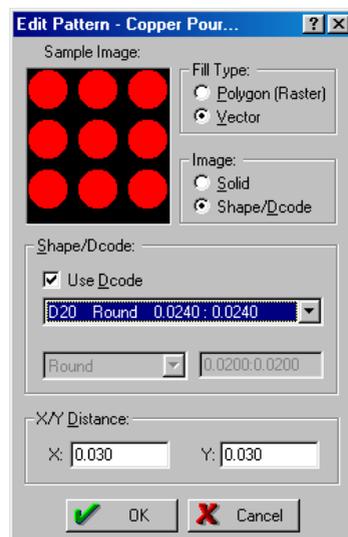


Рис. 7.58. Настройка шаблона рисунка заливки медью.

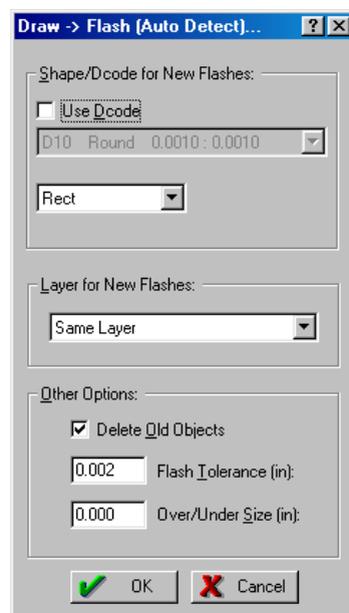


Рис. 7.59. Преобразование объектов топологии во флэши.

ется поле для задания ее размеров вручную. После выполнения всех необходимых настроек и нажатия кнопки **OK**, программа предложит указать область поиска похожих объектов. После щелчка правой кнопкой мыши программа автоматически найдет и заменит все объекты, аналогичные изначально выделенному.

Преобразование произвольного набора объектов в пользовательскую апертуру выполняется с помощью команды меню **Bonus » Draw to Custom Aperture**. После выделения нужных объектов на экране появится диалоговое окно, предлагающее ввести имя новой пользовательской апертуры.

Формирование отверстий

Преобразование флэшей в точки сверления выполняется с помощью команды меню **Tool » Create Drill**. Далее следует указать отдельные флэши круглой формы и завершить процедуру выделением щелчком правой кнопки мыши. По выбранным флэшам будет автоматически создана таблица инструментов, которая добавится к существующей в проекте, не затронув имеющейся в ней ранее информации. Отредактировать таблицу инструментов можно будет в любой момент времени в процессе работы над проектом. Просмотреть созданные отверстия сверления можно с помощью команды меню **View » Drill/Rout/Mill** — все отверстия будут показаны флэшами с наклонным крестиком.

Если требуется изменить диаметр инструмента, назначенный некоторым отверстиям, то следует выполнить команду меню **Tools » Modify Drill**. После выбора нужных флэшей появится диалоговое окно **Select Tool**, в котором из выпадающего списка надо выбрать нужный инструмент.

После того, как все отверстия сверления назначены, можно создать специальный слой сверления. Команда меню **Tools » Build Drill Layer (Plated-Through Hole)** работает аналогично команде **Create Drill**, за тем исключением, что вся информация об отверстиях сверления будет записана на новый слой **Drill Template**. Программа найдет все круглые флэши и перенесет их на этот слой, которому будет автоматически присвоен тип **Drill Top**. Напомним, что перед этим в таблице слоев должны быть определены слои **Top** и **Bottom**.

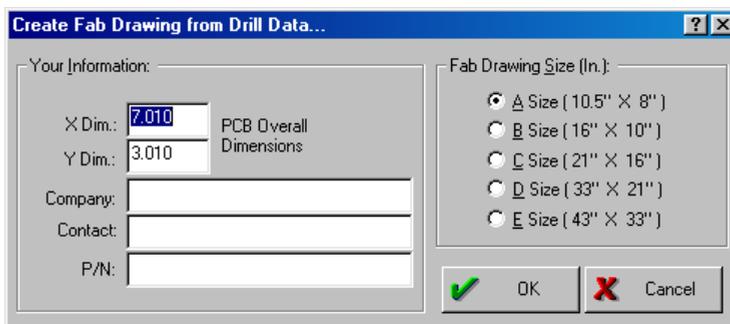


Рис. 7.60. Формирование чертежа сверления отверстий.

Как только слой **Drill** будет создан, пользователь может сформировать чертеж для сверления отверстий. Команда **Tool** » **Create Fab Drawing (from Drill)** откроет диалоговое окно, изображенное на рисунке 7.60. Программа автоматически определит внешние размеры платы и занесет данные в соответствующие поля **PCB Overall Dimensions**. Далее следует выбрать размер листа чертежа и заполнить графы основной надписи. После нажатия кнопки **OK** в проекте будет создан новый слой **Fab Layer**, на котором все отверстия, получаемые разными инструментами, будут показаны уникальными условными обозначениями. В дальнейшем этот слой может быть отредактирован аналогично любому другому слою проекта.

Векторная заливка контура

Любой замкнутый контур может быть залит штриховкой с помощью команды **Bonus** » **Fill Boundaries**. После выбора контура на экране появится диалоговое окно **Fill Boundaries**, показанное на рисунке 7.61. Здесь задаются следующие опции.

Layer: Определяет слой, на котором будет выполнена штриховка (этот же, новый или один из существующих).

Minimum Tool Size: Задаёт минимальную апертуру для прорисовки линий штриховки и, как следствие, шаг между ними.

Use Single Tool: Предписывает использовать только одну указанную апертуру. Выключение этой опции позволит использовать для штриховки все определенные в проекте апертуры, что позволит уменьшить объем файла.

Use Single Internal Offsets: Предписывает использовать только одно значение шага между линиями.

Use 'Snake' Pattern for Internal Area: Разрешает программе штриховать остатки контура "змейкой".

Delete Old Objects: Предписывает удалить изначальный контур заливки.

Генерация контуров объектов

В общем случае, когда необходимо просто просмотреть контуры объектов топологии, достаточно выключить опцию просмотра **View** » **Fill**. Если необходимо сгенерировать контуры объектов определенным кодом, то необходимо воспользоваться командой меню **Bonus** » **Generate Outlines**. После выбора нужных объектов появится диалоговое окно **Vector to Outline** (рис. 7.62). Здесь задаются следующие опции:

Trace Outline Mode: Один из трех стилей прорисовки контуров линий.

Layer: Определяет слой, на котором будет создан контур (этот же, новый или один из существующих).

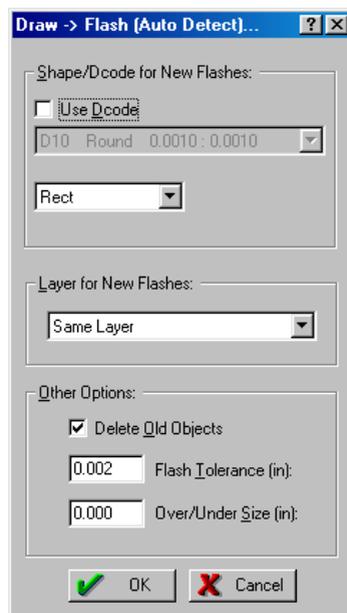


Рис. 7.61. Векторная заливка контура.

D-code: Задает D-код, с помощью которой будет прорисован контур объекта. Помимо определенных в таблице апертур D-кодов здесь имеется системная настройка по умолчанию 0.005 дюйма и возможность преобразования в растровый полигон.

Delete Old Objects: Предписывает удалить изначальный объект.

Удаление неиспользуемых контактных площадок

Команда меню **Tool** » **Pad Removal** запускает процедуру поиска и удаления всех неиспользуемых (изолированных) контактных площадок с внутренних слоев питания и заземления, позволяя избежать проблем при изготовлении платы. Разумеется, что для выполнения операции необходимо иметь в проекте хотя бы один слой типа Internal.

Панелизация печатной платы

Панелизация (мультиплицирование, размножение) печатной платы выполняется с помощью команды меню **Tools** » **Panelize PCB**. После выбора размножаемых объектов на экране появится диалоговое окно **Panelize PCB**, показанное на рисунке 7.63. Всем настройкам здесь присвоены значения по умолчанию, которые могут быть без труда изменены пользователем.

Image Size: Размер изображения по осям X и Y. Значения могут быть изменены, когда включена опция **Gap**.

Sheet Size: Размер листа пленки, на котором будет размножено изображение.

Offset: Минимальное значение шага между центрами размножаемых изображений. По умолчанию определяется, исходя из реальных размеров платы и минимального шага между ними.

Gap: Задает расстояние между изображениями плат по X и Y.

Minimum Border: Задает минимальное расстояние от изображения до границы листа пленки.

Part Count: Задает число размножаемых изображений на листе пленки, согласно заданному числу строк и столбцов при выключенной опции **Calculate**.

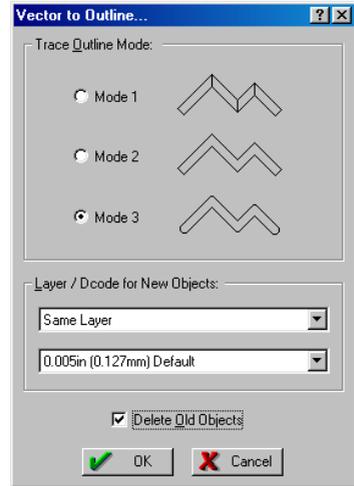


Рис. 7.62. Генерация внешнего контура объекта.

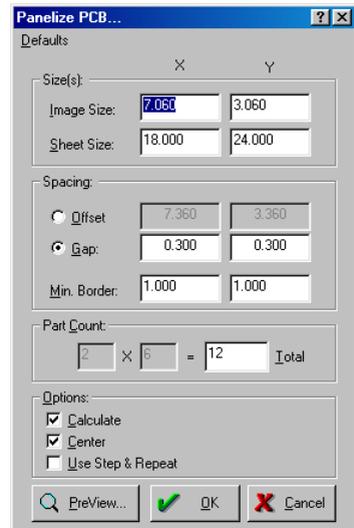


Рис. 7.63. Настройка функции панелизации платы.

Calculate: Включает режим расчета числа строк и столбцов размножаемого изображения, исходя из его размеров, зазоров и размеров листа пленки.

Center: Включает режим центрирования набора изображений платы на листе пленки.

Use Step & Repeat: включает возможность циклической генерации кодов для каждой копии изображения, что позволяет снизить объем файла проекта.

Preview: включает режим предварительного просмотра размноженных изображений.

Defaults: Предписывает использовать значения по умолчанию для **Sheet Size**, **Minimum Border** и **Gap**.

Фрезерование

Программа CAMtastic позволяет определять пути фрезерования. Путь фрезерования контура платы может быть получен автоматически с помощью команды **Tools » AutoRout PCB Border**. Сначала задается точка опускания инструмента, затем другие вершины контура и точка подъема инструмента. По окончании задания пути фрезерования появляется окно **Auto Route PCB** (рис. 7.64), где задается тип инструмента (с заранее заданными значениями диаметра, компенсации и глубины опускания) и допуски на опускание и подъем инструмента. После нажатия кнопки **OK** будет создан новый слой, на котором созданный путь фрезерования будет показан согласно настройкам меню **View » Drill/Rout/Mill**.

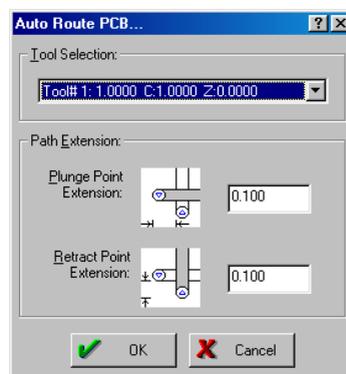


Рис. 7.64. Окно Auto Route PCB.

Программа CAMtastic дает возможность изменить ранее созданный или создать новый путь фрезерования, для чего используются команды **Tools » Modify Rout Path(s)** или **Tools » Create Rout Path(s)** соответственно.

Если контур фрезерования будет прорисован обычной ломаной линией, то для получения каждой из линий фреза будет опускаться в каждой начальной точке сегмента и подниматься в каждой конечной точке. Чтобы этого не происходило, рекомендуется преобразовывать простые ломаные линии в полилинии с помощью команды меню **Edit » Join**. Наоборот, если к полилинии требуется добавить маркеры опускания или подъема инструмента, то ее надо предварительно разбить и превратить в обычную ломаную линию с помощью команды **Edit » Break**.

Так как путь фрезерования, как правило, задается линиями, существует набор установок, определяющих порядок обработки встречающихся на пути флэшей, которые задаются в диалоговом окне **Create Rout Options**.

Drill Round Flashes: предписывает сверлить отверстия в местах, где на пути фрезерования встречаются круглые флэши.

Add Cleanup Cut: Включает алгоритм оптимизации пути фрезерования.

Plunge in Center: Задаёт режим опускания инструмента. При включенной опции инструмент опускается в центр апертур.

Заметим, что для фрезерования не могут быть использования апертур типа: **Custom**, **Diamond**, **Donut**, **Target** и **Thermal**.

Масштабирование слоев

Программа САМtastic допускает независимое масштабирование слоев проекта. После выполнения команды меню **Bonus** » **Scale Layers (X/Y)** появится окно, изображенное на рисунке 7.65. Здесь задается масштабный коэффициент для каждого отдельного слоя в процентах от изначального размера. Отрицательное число означает зеркальное отображение от соответствующей оси.

Заметим, что при таком масштабировании размеры апертур не меняются.

Изменение размеров объектов

Специальная функция программы САМtastic позволяет изменять размеры объектов (например, делать проводники толще или тоньше) без изменения масштаба изображения, что может быть полезно при прорисовке слоя шелкографии или трафаретов пайки волной. После выполнения команды меню **Tools** » **Spread/Reduce Sizes** и выделения нужных объектов на экране появится диалоговое окно **Spread/Reduce Sizes** (рис. 7.66).

В поле **Setting** задается значение в текущих единицах измерения, предписывающее на сдвинуть границу объекта в каждую сторону. Положительное значение приводит к расширению объекта, отрицательное — к сужению. Опция **Delete Old Object** разрешает удаление исходных объектов, а в выпадающем списке **Layer for New Objects** задается слой, на котором будут созданы новые объекты.

Формирование окон в трафарете для нанесения пасты

Преобразование двух любых контактных площадок в окна в трафарете для нанесения пасты для технологии поверхностного монтажа (SMT) выполняется с помощью команды меню **Tools** » **SMT Utilities** » **Homebase Pad Conversion**. Следует выбрать две контактные площадки и нажать правую кнопку мыши. Появится окно **Flash** → **Homebase Conversion**, показанное на рисунке 7.67, где задаются следующие опции.

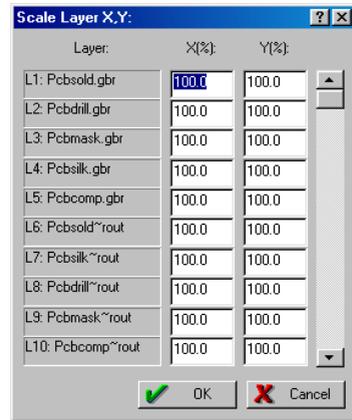


Рис. 7.65. Настройка масштабирования слоев.

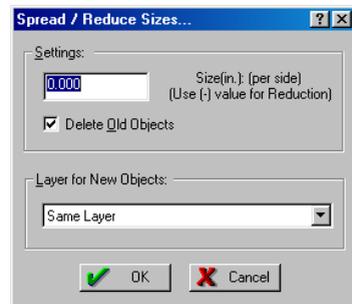


Рис. 7.66. Изменение размеров объектов.

Direction: Определяет направление скосов контактных площадок (внутри или наружу).

Edge Factor: Здесь задаются размеры скоса по горизонтали и вертикали в процентах от размера контактной площадки.

Delete Old Objects: Предписывает удалить изначальные объекты.

Layer: Задаёт слой, на котором будут созданы новые объекты.

Flash Tolerance: Задаёт допуск на поиск контактных площадок похожего размера в текущих единицах измерения.

После того, как все значения заданы, следует нажать кнопку **OK**, и программа снова перейдет в режим выделения. Окном охвата следует указать область поиска сходных по размеру контактных площадок. После щелчка правой кнопкой мыши программа выполнит преобразование всех найденных объектов в окна трафарета **Homebase Pads**.

Заметим, что новые объекты, не смотря на то, что не будут находиться в прямом контакте друг с другом, будут обрабатываться как единый объект. Другими словами, выбор одной контактной площадки автоматически влечет за собой выбор другой.

Формирование барьеров между контактными площадками

Формирование окна в трафарете для нанесения эпоксидной смолы, из которой получается барьер, предотвращающий растекание припоя и замыкание контактных площадок при технологии поверхностного монтажа (SMT) выполняется с помощью команды меню **Tools » SMT Utilities » Add Epoxy Bars**. Далее следует выбрать две контактные площадки и нажать правую кнопку мыши. Появится окно **Add Epoxy Bars**, показанное на рисунке 7.68, где задаются следующие опции.

Gap Factor: Здесь задаются размеры барьера по горизонтали и вертикали в процентах от размера контактной площадки.

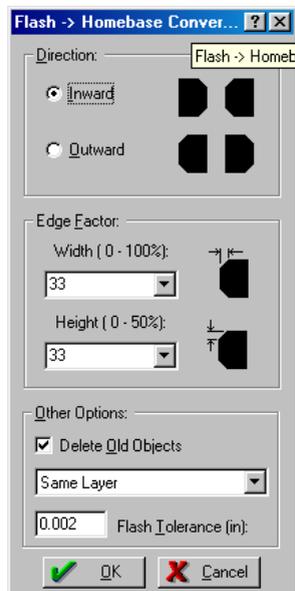


Рис. 7.67. Формирование трафарета для нанесения пасты.

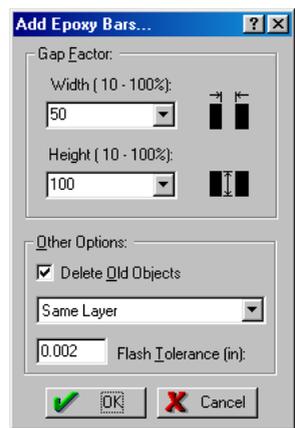


Рис. 7.68. Формирование защитных барьеров.

Delete Old Objects: Предписывает удалить изначальные объекты.

Layer: Задаёт слой, на котором будут созданы новые объекты.

Flash Tolerance: Задаёт допуск на поиск контактных площадок похожего размера в текущих единицах измерения.

После того, как все значения заданы, следует нажать кнопку **ОК**, и программа снова перейдет в режим выделения. Окном охвата следует указать область поиска сходных по размеру контактных площадок. После щелчка правой кнопкой мыши программа выполнит формирование окон для барьеров.

Отметим, что при повторном выполнении этой команды диалоговое окно **Add Epoxy Bars** появится только в случае выбора флэшей, размер которых не попадает в ранее заданные допуски.

Обработка циклических кодов (Step & Repeat)

Использование циклических кодов (Step & Repeat) позволяют значительно сократить размер выходного файла при выполнении таких операций, как панелизация (мультиплицирование) и вставка массивов объектов. В этом случае на экране будет отображаться только исходный левый нижний объект или группа объектов. Все копии, образованные с помощью кодов (Step & Repeat) будут показаны белыми прямоугольниками.

Такие массивы объектов могут быть изменены или разбиты на отдельные составляющие. Разбиение циклических кодов выполняется с помощью команды меню **Bonus » Step/Repeat » Explode S/R**. После щелчка левой кнопкой мыши на точке захвата исходного объекта белые прямоугольники исчезнут, а на их место будут вставлены копии размножаемого изображения.

Изменение параметров массивов **Step & Repeat** выполняется с помощью команды меню **Bonus » Step/Repeat » Modify S/R**. После щелчка левой кнопкой мыши на точке захвата исходного объекта на экране появится диалоговое окно **Modify Step & Repeat** (рис. 7.69), полностью аналогичное окну настройки вставки массива объектов.

С помощью команды меню **Bonus » Step/Repeat » Add Object to S/R** к размножаемому изображению могут быть новые объекты. Если новые объекты выступают за рамки ранее заданного контура изо-

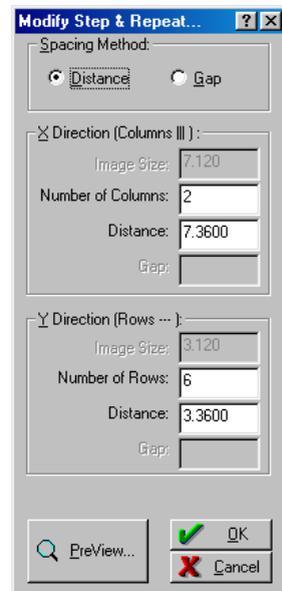


Рис. 7.69. Изменение параметров массивов Step & Repeat.

бражения, то это может привести к частичному наложению размножаемых копий. В этом случае необходимо откорректировать значения **Distance** или **Gap**.

Удаление объектов из размножаемого набора выполняется с помощью команды **Bonus** » **Step/Repeat** » **Remove Object from S/R**.

Отметим, что добавление или удаление объектов можно выполнять только с неразбитыми массивами. Если вы уже выполнили разбиение, например, для проверки правильности прорисовки массива, просто выполните операцию отката **Undo**.

Формирование каплевидных контактных площадок

Каплевидные контактные площадки позволяют гарантировать контакт в проводнике при смещении отверстия сверления. Их формирование выполняется с помощью команды меню **Tool** » **Teardrops**, после чего следует выбрать нужные объекты и нажать правую кнопку мыши. На экране появится окно **Teardrop** (рис. 7.70), в котором задаются следующие опции.

Pad: Добавляет к исходной площадке флэш вдвое меньшего размера.

Line: Добавляет к исходной площадке две линии.

Line Offset Factor: Задаёт относительный размер образующих "каплю" линий. Чем больше значение, тем длиннее "капля".

Нажатие кнопки **OK** завершает процедуру.

Корректировка слоя шелкографии

В ряде случаев требуется отредактировать слой шелкографии таким образом, чтобы находящиеся здесь объекты не попадали в окна в трафарете маски для пайки волной. Такая корректировка выполняется автоматически с помощью команды меню **Tools** » **Trim Silkscreen**. Откроется диалоговое окно, показанное на рисунке 7.71, где задаются следующие настройки.

Clearance: Задаёт зазор между краем контактной площадки и объектами на слое шелкографии.

Min. Segment Length: Задаёт минимальную длину

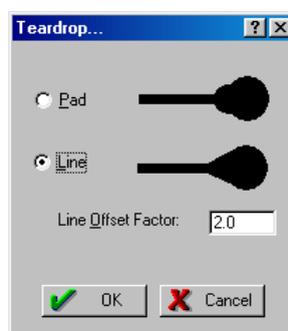


Рис. 7.70. Настройка формы каплевидных контактных площадок.

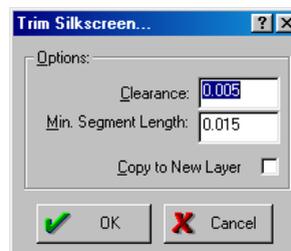


Рис. 7.71. Корректировка слоя шелкографии.

сегментов линий на слое шелкографии.

Copy to New Layer: При включении данной опции старый слой шелкографии будет скопирован с расширением **.OLD**, а на новый слой будут внесены изменения. В противном случае все изменения будут сделаны непосредственно на слое шелкографии. Разумеется, перед выполнением операции в проекте должны быть определены слои типа **Silk** и **Mask**.

Заливка неиспользуемых участков пленки

Заливка неиспользуемых участков панелизованной заготовки фотошаблона выполняется с помощью команды меню **Tools » Venting**. Откроется диалоговое окно, в котором следует задать заливаемые слои и рисунок заливки. Рисунок заливки задается аналогично описанной ранее операции заливки медью **Copper Pour**. Все имеющиеся в проекте сигнальные слои будут показаны и по умолчанию выделены в поле **Select Layer (for Venting)**. Панелизация платы может быть выполнена как обычным методом, так и с использованием циклических кодов **Step & Repeat**.

Литература

1. CAMtastic! 2000 Designers' Edition User's Guide. — San Diego: ACCEL Technologies, Inc., 2001.
2. Gerber RS-274X Format User's Guide. — South Windsor: Barco Gerber Systems Corp., 1998.
3. IBIS: I/O Buffer Information Specification Overview / W. Hobbs, A. Muranyi, R. Rosenbaum, D. Telian. — Intel Corp.
4. IBIS Forum I/O Buffer Modelling Cookbook. Revision 2.0X / Stephen Peters, Intel Corp., 1997.
5. Protel 99 SE Designers Handbook — Protel, 2000.
6. SPICE3 Version 3f3 User's Manual. T. Quarles, A.R. Newton, D.O. Pederson, A. San-giovanni-Vincentelli — University of California Berkeley, 1993.
7. View3D Users Manual — QualECAD, 2001.
8. XSPICE Simulator. Software User's Manual — Georgia Tech Research Corporation, 1992.
9. Александров К.К., Кузьмина Е.Г. Электротехнические чертежи и схемы. — М Энергоатомиздат, 1990.
10. Антонов А. П. Язык описания цифровых устройств AlteraHDL. Практический курс. — М.: "РадиоСофт", 2001.
11. Бибило П. Н. Основы языка VHDL. — М.: "Солон-Р", 2000.
12. Емец С. Verilog - инструмент ... — Схемотехника 1-4 , 2001.
13. Потапов Ю.В. Protel DXP для начинающих. — Компоненты и технологии №7, 2002.
14. Потапов Ю.В. Больше утилит, хороших и разных — PC WEEK/RE №2, 2001.
15. Потапов Ю.В. Изменение настроек среды проектирования Design Explorer — EDA Express №4, 2001.
16. Потапов Ю.В. Набор вспомогательных модулей производства Desktop EDA. — EDA Express №3, 2001.
17. Потапов Ю.В. Новые возможности Protel 99 SE — PC WEEK/RE №3, 2000.
18. Потапов Ю.В., Шалавин В.Л. Программирование на языке Client Basic. — EDA Express №4, 2001.
19. Разевиг В.Д. OrCAD 9.2 — М.: Солон-Р, 2001.
20. Разевиг В.Д. P-CAD 2000. Справочник команд. — М.: Горячая линия-Телеком. 2000.

21. Разевиг В.Д. Применение программ P-CAD и PSpice для схемотехнического моделирования на ПЭВМ. В 4 выпусках. — М.: "Радио и связь", 1992.
22. Разевиг В.Д. Проектирование печатных плат в P-CAD 2001. — М.: Солон-Р, 2001.
23. Разевиг В.Д. Система P-CAD 8.5-8.7: Руководство пользователя. — М.: Солон-Р, 1999.
24. Разевиг В.Д. Система проектирования OrCAD 9. — М.: Солон-Р, 2001.
25. Разевиг В.Д. Система проектирования печатных плат ACCEL EDA 12.1 (P-CAD для Windows). — М.: СК Пресс, 1997.
26. Разевиг В.Д. Система проектирования печатных плат ACCEL EDA 15 (P-CAD 2000). — М.: Солон-Р, 2000.
27. Разевиг В.Д. Система проектирования цифровых устройств OrCAD. — М.: Солон-Р, 2000.
28. Разевиг В.Д. Система сквозного проектирования электронных устройств DesignLab 8.0. — М.: "Солон", 1999.
29. Разевиг В.Д. Система схемотехнического моделирования и проектирования печатных плат Design Center (Pspice). — М.: СК Пресс, 1996.
30. Разевиг В.Д. Средства подготовки фотошаблонов печатных плат CAMtastic! 2000 Designers' Edition. — EDA Express №3, 2001.
31. Разевиг В.Д., Блохнин С.М. Система P-CAD 8.5. Руководство пользователя. — М.: ООО "Илекса", 1996.
32. Разработка и оформление конструкторской документации РЭА. / Под. ред. Романычевой Э.Т. — М.: Радио и связь, 1989.
33. Справочник по конструированию радиоэлектронной аппаратуры (печатные узлы). — Киев: Техника, 1985.
34. Стешенко В.Б. ACCEL EDA. Технология проектирования печатных плат. — М.: Нолидж, 2000.
35. Сучков Д.И. Адаптация САПР P-CAD к отечественному технологическому оборудованию. — Обнинск: Призма, 1993.
36. Сучков Д.И. Основы проектирования печатных плат в САПР P-CAD 4.5, P-CAD 8.5-8.7 и ACCEL EDA. — М.: "Горячая линия-Телеком". 2000.
37. Сучков Д.И. Проектирование печатных плат в САПР P-CAD 4.5, P-CAD 8.5 и ACCEL EDA. — М.: Малип, 1997.
38. Сучков Д.И. Проектирование печатных плат в САПР P-CAD 4.5: Учебно-методическое пособие. — Обнинск: Микрос, 1992.
39. Гархов А.С. И, наконец, он появился — Protel DXP компании Altium. EDA Express №6 2002.

40. Уваров А. P-CAD 2000, ACCEL EDA. Конструирование печатных плат. Учебный курс. — СПб.:Питер, 2001.

Предметный указатель

.	
.IC	248
.NS	247
A	
Absolute ABS	412
ABSTOL	301
AC анализ	
настройка и запуск	223
ACCT	304
ADCSTEP	306
Add Epoxy Bars	
в CAMtastic	782
Add Object to S/R	
в CAMtastic	783
Add Template to Clipboard	83
Align Layers	
в CAMtastic	772
annotation	
forward	169
ANSI	
title block	86
границы лиситов	86
Apertures Report	
в CAMtastic	758
Arc	
в CAMtastic	765
Array	
в CAMtastic	768
ASCII	
Hex	411
ASCII	
HL	411
Assembly Utilities	
в CAMtastic	773
Assign Ref Designators	
в CAMtastic	773
attributes to match by	533
Auto Create Composite	
в CAMtastic	774
Auto Zoom	83
auto-junction	
включение / выключение	82
Automatically Remove Loops	507
AUTOPARTIAL	305
AutoRout PCB Border	
в CAMtastic	780
avoid obstacle	507
B	
BADMOS3	303
Berkeley PLA	411
Bill of Materials	
в редакторе схем	181
BJT	
моделирование	279
BOM	
настройка	700
BOOLH	303
BOOLL	303
BOOLT	303
Break	
в CAMtastic	769
Build Composite	
в CAMtastic	774
bus	
обеспечение связанности	122
обзор	122
bus entry	
обзор	122
BYPASS	304
C	
CAM Manager	
генерация Gerber файлов	697
генерация отчета о контрольных	
точках	702
генерация отчета о проверке правил	
проектирования	703
генерация перечней используемых	
материалов (BOM)	700
генерация файлов для оборудования	
размещения (Pick and Place) ...	701
генерация файлов сверления	700
использование	693
Center of Object	83

Chamfer		
в CAMtastic.....	769	
change scope.....	534	
CHGTOL.....	301	
Circle		
в CAMtastic.....	766	
clearance constraint.....	597	
Clipboard Reference.....	83	
Close Composite		
в CAMtastic.....	775	
component		
selecting a PCB component from the		
schematic.....	625	
Component Drag.....	508	
Composite Layers		
в CAMtastic.....	774	
confirm global edit.....	506	
CONVABSSSTEP.....	304	
convert special strings.....	508	
Convert Special Strings.....	83	
CONVLIMIT.....	304	
CONVSTEP.....	304	
Coordinate Box		
в CAMtastic.....	727	
Copper Area Calculation		
в CAMtastic.....	758	
Copper Pour		
в CAMtastic.....	775	
copper trace layers.....	502	
Copy		
в CAMtastic.....	769	
copy attributes.....	534	
Copy to Layers		
в CAMtastic.....	769	
Create & Group Parts		
в CAMtastic.....	773	
Create Drill		
в CAMtastic.....	777	
Create Fab Drawing (from Drill)		
в CAMtastic.....	778	
Create Rout Path(s)		
в CAMtastic.....	780	
cross probe		
горячая связь между схемой и		
печатной платой.....	626	
Cross Probing.....	80	
CUPL		
синтаксис.....	390, 396	
CURRENTMNS.....	305	
CURRENTMXS.....	305	
Custom Aperture Library File		
в CAMtastic.....	759	
D		
DC анализ		
настройка и запуск.....	225	
проблемы.....	298	
DCode/layer Report		
в CAMtastic.....	760	
deactivate unused OR terms.....	407	
decals		
задание топологического		
посадочного места в редакторе		
схем.....	132	
по умолчанию.....	141	
DEFAD.....	301	
DEFAS.....	301	
Default Power Object Names.....	82	
Default Primitives.....	84	
Default Template File.....	83	
DEFL.....	301	
DEFW.....	301	
DeMorgan.....	407	
Design Database.....	14	
Design Explorer		
изменение настроек.....	38	
обзор.....	35	
designator		
в редакторе схем.....	133	
device		
secure.....	407	
secure PLD device.....	407	
Digital		
SimCode simulation language.....	307	
Dimension Setting		
в CAMtastic.....	767	
display mode.....	510	
Display Printer Fonts.....	83	
DM-PL		
вывод на печать.....	693	
draft thresholds.....	509	
Drag Orthogonal.....	82	
Draw to Flash		
в CAMtastic.....	776	
DRC		
автоматическая проверка.....	505	

интерактивная проверка.....	673
настройка генерации отчета в CAM Manager.....	703
настройка пакетной проверки.....	673
обзор.....	672
отображение ошибок.....	504
отчет.....	674
устранение нарушений.....	675
DRC/DFM Check в CAMtastic.....	756
DRC/DFM Errors в CAMtastic.....	760
DRC/DFM View в CAMtastic.....	726
drill drawing.....	503
drill guide.....	503
DRIVEMNS.....	305
DRIVEMXS.....	305
DRVMNTYMX.....	306
DXF/DWG schematic import/export.....	195
импорт и экспорт в редакторе плат	718
E	
EDIF.....	412
Electrical Grid.....	86
Electrical Rule Check настройка.....	158
Ellipse в CAMtastic.....	766
Erase в CAMtastic.....	769
ERC настройка.....	158
error list LST.....	411
expanded macro MX.....	411
Explode в CAMtastic.....	770
Explode Composite в CAMtastic.....	775
Explode S/R в CAMtastic.....	783
Extend в CAMtastic.....	770
Extend Selection.....	505
F	
FIELD ключевые слова ПЛИС.....	395
Fill Boundaries в CAMtastic.....	778
Film Box в CAMtastic.....	754
Flash в CAMtastic.....	764
footprint задание топологического посадочного места в редакторе схем.....	132
по умолчанию.....	141
frequency and phase, viewing.....	254
From-To использование в правилах проектирования.....	618
обзор.....	638
создание.....	638
G	
Generate Outlines в CAMtastic.....	778
Gerber апертуры.....	706
генерация файлов.....	693
общие сведения.....	705
Gerber файлы настройка.....	697
GMIN.....	301
GMINSTEP.....	304
Graphical Editing.....	83
H	
hidden pin создание связанности.....	148
Hidden Pins отображение на схемах.....	134
highlight in full.....	509
Homebase Pad Conversion в CAMtastic.....	781
Horizontal в CAMtastic.....	768
Hot Spot	86
HP-GL вывод на печать.....	693

Netlist Extraction	
в CAMtastic.....	756
NetList View	
в CAMtastic.....	726
NOOPALTER	304
NOOPITER	304
O	
Object's Electrical Hot Spot	83
Offset	
в CAMtastic.....	771
OLE сервер	
обзор	37
One-hot-bit state machines.....	408
online DRC	505
OPTS	304
OrCAD	
импорт из OrCAD Capture... 191, 193	
импорт проектов плат.....	719
OrCAD Load Options.....	82
Ordinate	
в CAMtastic.....	768
organization	
добавление на схему информации о	
разработчике	87
Organization	
вкладка.....	87
P	
Pad Removal	
в CAMtastic.....	779
Palasm PDS	411
Panelize PCB	
в CAMtastic.....	779
part type	
в редакторе схем	133
paste mask	503
pattern	
задание топологического	
посадочного места в редакторе	
схем	132
по умолчанию	141
PCB Sale Quote	
в CAMtastic.....	761
PDF PDF	412
pick and place	
отчет.....	701
pin	
обзор	124
Pin Options	82
PIVREL	302
PIVTOL	302
Plow Through Polygons	508
Point to Point	
в CAMtastic	755
Polygon	
в CAMtastic	766
Polyline	
в CAMtastic	764
port	76
cross referencing.....	149
обзор	123
создание связанности	148
Ports Only Global.....	150
Postscript	
печать чертежей.....	708
Postscript полезные советы	708
power port	76
обзор	123
создание связанности	148
process container	
обзор	126
PROPMNS	305
PROPMXS	305
Protect Locked Objects.....	506
push and shove	507
push obstacle	507
Q	
Query	
в CAMtastic	755
Query Group	
в CAMtastic	756
Query Net	
в CAMtastic	756
Quote	
в CAMtastic	761
R	
Radius	
в CAMtastic	768
RAMP TIME	304
Rectangle	
в CAMtastic	765
Redo	
в редакторе плат	546

в редакторе схем	120	shift step size	507
redraw layers.....	509	Show	509
RELTOL	302	Show Borders	86
Remove Duplicates.....	506	show/hide	510
Remove Object from S/R		signal integrity	
в CAMtastic.....	784	checking the rules	677
Reports		signal layers.....	502
в CAMtastic.....	758	SimCode	
resolution		компиляция модели.....	311
design database.....	492	определение языка.....	314
room		синтаксис языка.....	317
работа с областями размещения .	627	Simlist.TXT файл	311
Rotate		simulation	
в CAMtastic.....	772	viewing frequency and phase	254
Rout/Mill отчет		SIMWARN.....	306
в CAMtastic.....	763	Single '\ Negation	83
RSHUNT	306	single layer mode.....	509
S		singular matrix error	
Scale		устранение.....	306
в CAMtastic.....	772	slider hand	543
Scale Layers (X/Y)		SmartDoc	
в CAMtastic.....	781	обзор	37
Schematic Editor		SmartTeam	
обзор	70	обзор	37
характеристики	70	SmartTool	
secure device.....	407	обзор	34
selection		Snap	
selecting a PCB component from the		в CAMtastic	751
schematic	625	Snap Grid.....	86
share product terms	407, 408	Snap to Center	505
sheet		solder mask.....	503
schematic title block	86	special strings	
sheet entry.....	76	использование на схемах	87
обзор	123	отображение содержимого	83
создание связанности	148	SPICE	
Sheet Options.....	84	Gmin stepping failed	296
Sheet Part Filename	141	singular matrix.....	296
sheet path		source stepping failed	296
в редакторе схем	133	добавление информации в список	
sheet symbol		соединений.....	246
обзор	123	задание переменных	299
Sheet Symbols/Port Connections.....	153	множители.....	297
sheet template		ошибки.....	299
создание.....	88	предупреждения	299
sheet templates.....	87	проблемы сходимости.....	295
sheet units	87	SPICE, совместимость.....	212
		Spread/Reduce Sizes	

в CAMtastic.....	781	максимальное число команд в редакторе плат	506
SRCSTEP	304	максимальное число команд в редактре схем.....	84
Status отчет		Undo Stack Size в редакторе схем.....	84
в CAMtastic.....	763	union	
Step & Repeat		группы компонентов на платах ..	626
в CAMtastic.....	783	URC линии передачи.....	287
step size	507	Use DeMorgan.....	407
STEP_OFF	317	use net color for highlight.....	509
STEP_ON	317	V	
suppress product terms merging	408	Venting	
T		в CAMtastic	785
Teardrop		Vertex	
в CAMtastic.....	784	в CAMtastic	772
TEMP	303	Vertical	
Template	86	в CAMtastic	768
Text		Visible Grid	86
в CAMtastic.....	767, 772	VNTOL.....	303
timestep too small.....	298	W	
изменить шаг	302	Windows	
title block		драйверы для плоттеров.....	709
добавление на схему информации о разработчике	87	wire	
Title block.....	86	обзор	122
TNOM	303	X	
Toggle		X/Y Coordinate List	
в CAMtastic.....	753	в CAMtastic	763
TPMNTYMX	306	XNF.....	411
TRANMNS.....	305	XOR вентили	407
TRANMXS.....	305	XSpice.....	307
transient analysis		A	
настройка и запуск.....	220	абсолютное начало координат на чертежах плат	497
Translucent		автоматический масштаб при печати схем.....	180
в CAMtastic.....	753	автоматическое панорамирование	
transparent layers.....	509	настройка в редакторе плат	506
Trim Silkscreen		автоматическое соединение	
в CAMtastic.....	784	включение / выключение	82
TRTOL.....	303	авторазмещение на схемах	95
TRYTOCOMPACT	303	автосоединение проводников на схемах	96
TTMNTYMX	306	автотрассировка	
TURBO биты	394		
U			
Undo			
в редакторе плат.....	546		
в редакторе схем	120		

блокирование ранее разведенных проводников	661	добавление к списку используемых библиотек	130
настройка	661	закружка в редактор схем	130
области	665	неизменяемые текстовые поля ...	141
опции	664	о компонентах	75
связей компонента	665	подключение библиотек посадочных мест	565
цепи	664	поле sheet path	155
Адрес подчиненного листа	141	работа со старыми библиотеками	32
анализ		редактирование библиотек посадочных мест	574
настройка программы моделирования	218	редактирование библиотек элементов схем	136
анализ передаточных функций		редактора принципиальных схем	128
настройка и запуск	239	редактора печатных плат	565
анализ переходных процессов		создание библиотеки проекта	577
настройка и запуск	220	создание новой библиотеки элементов схем	136
проблемы	298	библиотеки элементов принципиальных схем	128
анализ целостности сигналов		библиотечные поля компонентов ..	134
настройка	676	биполярные транзисторы моделирование	279
правила проектирования	610	битовые поля проектирование ПЛИС	395
просмотр формы сигналов	677	блок процедуры обзор	126
анализ шумов		блокирование блокирование документов проекта	30
настройка и запуск	240	блокирование ранее разведенных проводников	661
аналоговое моделирование		булевы выражения	303
определение	212	буфер обмена в редакторе печатных плат	528
аннотирование		в редакторе принципиальных схем	103
обратное	710	добавление шаблона к содержимому	83
апертуры		быстрое копирование объектов на схемах	109
в Gerber файлах	705	В	
загрузка и редактирование	707	векторные фотопроттеры	704
обзор	706	верификация	
Апертуры			
в САМtastic	746		
асинхронные векторы			
моделирование ПЛИС	423		
Б			
база данных проекта	14		
барьеры			
в САМtastic	782		
библиотеки			
где находятся библиотеки посадочных мест	565		
где хранятся библиотеки элементов схем	129		
группы компонентов	142		

в САМtastic.....	755	отображение точек данных.....	255
принципиальной схемы.....	158	просмотр на одном графике	252
верификация прокта		время установления	
в редакторе печатных плат.....	672	моделирование.....	305
верификация схем.....	158	переменная перекрытия	306
вернуть		вспомогательные переменные	
в редакторе плат.....	546	проектирование ПЛИС	391
в редакторе схем	120	вставка	
вершины		массивов объектов в редакторе плат	
в САМtastic.....	772	530
добавление к полилинии	100	массивов объектов на схемы	105
удаление.....	100	специальная в редакторе плат	529
видимая сетка		вход листа	
редактор принципиальных схем... 86		обзор	123
видимые сетки		вход шины	
в редакторе печатных плат.....	499	обзор	122
отображение	504	входная нагрузка	
виртуальные линии связи.....	504	моделирование.....	305
виртуальные проекта на ПЛИС	438	переменная перекрытия	306
виртуальные устройства.....	406	входы листа	76
Влаженные команды		связанность	78
в редакторе принципиальных схем		создание связанности	148
.....	116	вывод	
вложенные функции редактирования		обзор	124
.....	541	вывод чертежей плат на печать.....	693
внешние базы данных		выводы	
связь из редактора схем.....	183	определение выводов ПЛИС	394
внутренние слои питания		отображение скрытых выводов на	
разделение	656	схемах	134
внутренние слои питания		проектирование ПЛИС	391
использование	655	размещение на элементах схем ..	140
назначение цепи	655	расположение названий	82
просмотр.....	656	связанность	78
соединение с переходными		список выводов компонентов.....	680
отверстиями.....	656	стыковка с внутренней схемой	
список используемых	681	ПЛИС.....	387
временные диаграммы		выделение	
просмотр на отдельных графиках		в редакторе печатных плат	520
.....	252	добавление объектов	505
временные диаграммы		из панели редактора плат.....	526
идентификация при монохромной		компонента на плате из схемы ..	526
печати.....	255	мастер Selection Wizard.....	528
использование измерительных		объектов на схемах	101
маркеров	256	подсвечивание	509
масштабирование.....	250	посвечивание цепей.....	509
отображение сигналов.....	249		

работа с выделенными объектами	103	идентификация объектов на схемах	107
редактор принципиальных схем...98		использование символов-заменителей на схемах	108
создание массивов объектов на платах	530	использование фигурных скобок	108
создание массивов объектов на схема	105	копируемые атрибуты	107
стратегия выделения на схемах ..	101	область действия на схемах	107
Выделение объектов		примеры плат	534
в САМtastic	731	редактор принципиальных схем	105
выделение объектов в редакторе плат	522	глухие и скрытые переходные отверстия	552
выделение фокусом		горячая связь	
редактор принципиальных схем...98		схемой и печатной платой	626
выделенные объекты		Горячая связь	80
переход к нужному	518	горячие клавиши	
выравнивание		в редакторе плат	541
компонентов на платах	628	в редакторе схем	117
объектов на схемах	111	выделение	102
выходные файлы		выделение в редакторе плат	523
генерация	693	наиболее часто используемые в редакторе схем	119
выходные форматы		обзор	38
Absolute ABS	412	перетаскивание объектов на схемах	113
ASCII HL	411	размещения компонентов	545
Berkely PLA	411	редактор принципиальных схем	116
error list LST	411	трассировка	649
expanded macro MX	411	Горячие клавиши	
Hex	411	в САМtastic	729
Jedec	411	горячие клавиши редактора печатных плат	543
Palasm PDS	411	границы	
PDIF PDF	412	листов принципиальных схем	85
XNF	411	графическое изображение	
проектирование ПЛИС	410	добавление на схему	125
выходные форматы		графическое редактирование	
EDIF	412	объектов на платах	520
Г		объектов на схемах	100
генерация выходных файлов	693	групповая разработка	
Генерация контуров		настройка прав доступа	24
в САМtastic	778	создание члена группы разработчиков	25
главный лист	151	групповые объекты	
обзор	145, 146	в редакторе печатных плат	559
глобальное редактирование		группы	
в редакторе печатных плат	532	группы компонентов	142
замена части строки	108		

Д

диагональная трассировка	666
диапазон адресов проектирование ПЛИС	395
диод моделирование	278
директивы	
обзор	76
топологические	80
добавление листа схемы в проект ..	147
добавление слоев на плату	579
документ	
блокирование	30
импорт	21
копирование	23
перемещение	23
связывание	22
удаление	23
экспорт	22
драйверы	
для плоттеров	709
дуги	
на платах	555
размещение на чертежах плат ..	555
редактирование на платах	556
Дуги в САМtastic	765
дюймовая система переключение в редакторе плат ..	497

Е

единицы измерения	
в редакторе принципиальных схем	87
переключение	497
разрешение на чертежах плат	497

З

загружаемые файлы	
проектирование ПЛИС	411
загрузка списков соединений	713
заданные выражением источники ..	272
задержка распространения моделирование	305
переменная перекрытия	306
зазоры	597

Заливка

в САМtastic	775
заливка контура	
в САМtastic	778
запрет перемещения заблокированных объектов	506
Запросы	
в САМtastic	755
Зеркальное отображение	
в САМtastic	771

И

трассировка	652
идентификаторы цепей	
обзор	76
область действия	148
идентификация цепей на платах ..	642
иерархия	
выбор модели	152
инструменты для построения	157
обзор	145
преобразование сложной в простую	154
простая	151
работа с иерархическим проектом	156
сложная	153
изгибы проводников	549
изменение	
объектов на платах	519
изменение параметров элементов	
настройка и запуск	233
изменение температуры	
настройка и запуск	236
изменяющиеся маркеры	
использование	256
изменяющиеся параметры	
интерпретация результатов	256
измерение расстояний в редакторе плат	540
Измерения	
в САМtastic	755
импедансы цепей	611
импорт	
в базе данных проекта	21
Импорт DXF и DWG файлов	

в САМtastic.....	734	выделять отдельные объекты на	
Импорт Gerber файлов		схемах	102
в САМtastic.....	737	найти и заменить текст на схеме	109
Импорт HPGL/HPGL2		разбиение проводников на схема	100
в САМtastic.....	739	размещать проводники.....	94
Импорт NC Drill файлов		удалить излом проводника на схеме	
в САМtastic.....	734	100
Импорт NC Mill/Route		Каплевидные контактные площадки	
в САМtastic.....	739	в САМtastic	784
Импорт файлов		каплевидные контактные площадки и	
в САМtastic.....	733	переходные отверстия.....	671
Импорт файлов апертур		катушки индуктивности	
в САМtastic.....	740	моделирование связанных катушек	
имя компонента в редакторе схем..	132	246
индуктивность		кварцевые резонаторы	
моделирование	278	моделирование.....	285
Инсталляция системы Protel 99 SE		клавиши	
процесс инсталляции	5	горячие в редакторе схем.....	117
интерактивная трассировка.....	507	классы	
интерактивное размещение		создание из выделения.....	527
компонентов	628	создание классов объектов на	
исправление ошибок		платах	595
на схемах	163	создание на основе информации из	
источники		схемы	625
заданные выражением	272	коды черчения	
использование в моделировании	257	общие сведения.....	705
кусочно-линейного сигнала	262	комментарии	
линейные зависимые	268	в редакторе плат	569
нелинейные зависимые	272	компилятор ПЛИС	
периодических импульсов	260	синтаксис исходного файла	390
постоянные	258	компиляция	
синусоидальные	259	проектирование ПЛИС	403
частотно-модулированных сигналов		компиляция SimCode модели	311
.....	266	Композитные слои	
экспоненциальные	264	в САМtastic	774
итерации		компоненты	
увеличение числа	297	атрибуты в редакторе схем.....	132
К		атрибуты посадочных мест.....	568
как		в редакторе печатных плат	565
выполнить глобальное		включение проводников в	
редактировании платы.....	534	посадочное место.....	572
создать пользовательский шаблон	88	выравнивание на платах.....	628
Как		где находятся библиотеки	
выделять объекты в окне.....	102	посадочных мест.....	565
		где хранятся элементы схем	129
		группирование	142

группы (union) на платах.....	626	список используемых.....	681
группы компонентов	142	текстовые поля.....	134
для моделирования	274	тип элемента	133
доступ к библиотека в редакторе		топологическое посадочное место	
плат	565	132
доступ к элементам схем.....	130	топологическое посадочное место	
задание топологического		по умолчанию	141
посадочного места в редакторе		элементы принципиальных схем	128
схем	132	конденсатор	
изменение назначенного		моделирование.....	277
посадочного места	571	конечные автоматы	
изменение цвета	135	проектирование ПЛИС	400
комментарии в редакторе плат ...	569	контактные площадки	
копирование	142	минимальное расстояние до изгиба	
копирование из одной библиотеки в		601
другую.....	576	минимальное расстояние до	
копирование с платы в библиотеку		переходного отверстия.....	602
.....	572	обзор	550
модификация на печатной плате	571	обозначения.....	551
обзор	75	отображение имен цепей	509
обновление посадочных мест	576	отображение номеров.....	509
обозначение в редакторе плат ...	569	отображение отверстий.....	504
ориентация	134	переход к нужной	518
перенос на другой слой	623	размещения	551
переход к нужному	517	редактирование.....	552
поворот	623	скачок ширины проводника	601
подготовка для моделирования ..	289	контрольные точки	
позиционирование на платах	624	автоматическое добавление.....	670
поиск в библиотеках	131	добавление на платы	667
поиск в библиотеках посадочных		использование.....	606
мест	567	настройка генерации отчета	702
поиск моделей	275	обзор	667
поле part type	133	отчет о местоположении.....	670
поле sheet path	155	поиск	669
разгруппировка в редакторе плат		стиль	605
.....	572	удаление	671
размещение.....	545	координатные метки.....	616
размещение компонентов в сетку		координатные метки	
.....	647	на чертежах плат.....	564
размещение на платах	622	координаты	
редактирование контактных		система координат редактора	
площадок	552	печатных плат	497
создание библиотеки проекта.....	577	копирование	
создание посадочного места	575	быстрое на схемах	109
создания элемента схемы.....	139	из редактора плат в буфер обмена	
сопоставление схемы и платы	174	Windows.....	690

компонентов в библиотеках.....	142	маркер-манипулятор	100
точка привязки	83	маркеры ошибок	
копоненты		переход к нужному.....	518
sheet path	133	маршруты	
адрес подчиненного листа	133	обзор	638
курсор		создание.....	638
стиль.....	506	маска	
кусочно-линейного сигнала,		в SAMtastic	781
источники	262	маски	
Л		для пайки и насесения пасты.....	503
линейные зависимые источники	268	маски для нанесения паяльной пасты	
линии		503
обзор	546	маски для пайки волной.....	503
размещение.....	547	массивы объектов	
Линии		вставка на схемы.....	105
в SAMtastic.....	764	размещение в редакторе плат	530
линии передачи		мастер	
моделирование	286	топологических посадочных мест	
линии передачи без потерь	286	575
линии связи		мастер выделения в редакторе плат	528
отображение	640	мастера	
линии связи		создания печатных плат	582
отображение	504	масштаб	
линии штриховки полигонов	561	просмотра печатной платы	514
линия передачи с потерями.....	287	Масштабирование	
листы		в SAMtastic	772
добавление подчиненного листа	147	масштабирование кривых	250
размеры листов принципиальных		Масштабирование слоев	
схем	84	в SAMtastic	781
связи между листами схем.....	147	математическая обработка сигналов	
листы принципиальных схем		243
границы.....	85, 86	межлистовые соединения	147
логическая связанность		меню команд	
обзор	77	обзор	38
логические выражения		метка цепи	
проектирование ПЛИС.....	391	обзор	122
М		метки цепей.....	76
макросы	60	инверсия	148
максимальные размеры плат	497	связанность	78
максимум совместного использования		создание связанности	148
термов производства	409	метод интегрирования	
малосигнальный анализ		задание.....	300
настройка и запуск.....	223	проблемы.....	299
маркер начала координат	509	методы минимизации	409
		метрическая системы	
		переключение в редакторе плат .	497
		механические слои	502

миллиметры	
переключение размерности в редакторе плат.....	497
милы	
переключение размерности в редакторе плат.....	497
минимальная нагрузка по выходу	
моделирование.....	305
минимизация	
выключить.....	409
сравнение методов.....	409
минимизация выходов.....	408
мнемоническое обозначение устройств.....	405
многолистовые проекты	
методы построения.....	149
обзор.....	144
управление.....	145
множители	
моделирование.....	215
проблемы.....	297
модели	
в программе моделирования.....	274
моделирование	
компоненты.....	274
моделирование	
виртуальных проектов на ПЛИС	438
добавление информации в список соединений.....	246
интерпретация результатов при изменяющихся параметрах.....	256
использование измерительных маркеров.....	256
источники.....	257
источники напряжения.....	257
источники тока.....	257
компиляция SimCode модели.....	311
масштабирующие множители.....	215
математическая обработка сигналов.....	243
метод интегрирования.....	299
модели.....	274
начало работы.....	214
неизменяемые библиотечные поля компонентов.....	289
неисправностей в ПЛИС.....	439
обработка ошибок.....	295
ограничения.....	213
определение узлов.....	218
определение языка SimCode.....	314
отображение точек данных.....	255
отображение формы сигнала.....	249
ошибки.....	299
переменные SPICE.....	299
подготовка и запуск.....	218
поддержка моделей производителей.....	213
представление чисел.....	215
предупреждения.....	299
пример SimCode.....	311
проблемы сходимости.....	295
проектов на ПЛИС.....	413
просмотр результатов.....	248
создание компонентов.....	289
создание нового SimCode устройства.....	308
файл ABS проектов на ПЛИС.....	412
цифровые компоненты.....	288
цифровые схемы.....	307
моделирование неисправностей	
проектирование ПЛИС.....	439
модульные проекты.....	154
МОП-транзистор	
моделирование.....	281
морфинг.....	109
Н	
нагрузка	
моделирование.....	305
нагрузка по выходу	
переменная перекрытия.....	306
настройка	
AC анализ.....	223
DC анализ.....	225
Design Explorer.....	38
Monte Carlo анализ.....	228
анализ передаточных функций.....	239
анализ переходных процессов.....	220
анализ шумов.....	240
выбор типа анализа.....	218
изменение параметров элементов.....	233

изменение температуры	236
принтера	180
расчет комплексных сопротивлений	241
расчет рабочей точки.....	227
среда проектирования	38
статистический анализ	228
Фурье анализ	237
настройки	
вкладка Layers	500
редактор принципиальных схем...82	
редактора печатных плат	504
настройки слоев	500
начало координат	
абсолютное на чертежах плат.....	497
переход в абсолютное	517
переход в относительное.....	517
текущее	497
начальные условия.....	248
задание.....	222, 247, 248
использование	248
неизменяемые библиотечные поля	
компонентов.....	289
неиспользуемые участки	
металлизации	561
нелинейные зависимые источники	272
О	
области заливки	
размещение.....	554
редактирование	554
области металлизации	
создание	659
области размещения	
работа с ними	627
создание на основе информации из	625
схемы	625
область действия	
идентификаторы цепей.....	148
область действия идентификаторов	
цепей	
Net Labels and Ports Global	151
Ports Only Global	150
Sheet Symbols/Port Connections...153	
настройки для формирования	
списка соединений.....	196
область действия правил	
проектирования	
символы-заменители	589
стратегия задания	589
область действия правила	
проектирования	586
обновление	
схемы после изменений на плате... 711	
элементов на схемах.....	143
обновление платы	
после изменения схемы.....	169
обновление схемы	
по изменениям в плате	710
обновление экрана	
отмена	541
прерывание в редакторе	
принципиальных схем.....	116
обозначение	
по умолчанию	141
обозначения	
в редакторе плат	569
контактные площадки	551
присвоение в редакторе плат.....	710
обратное аннотирование	710
трассировка	652
Объединение линий	
в САМtastic	770
объекты	
выделение на схемах	101
добавление к выделению на платах	522
добавление к выделению на схемах	101
комплексное выделение в редакторе	522
плат	522
размещение на схемах.....	94
редактирование в редакторе плат	519
редактирование на схемах	97
редактора печатных плат	546
редактора принципиальных схем	121
удержание при перемещении	83
окно проекта	
изменение конфигурации	13
окно просмотра результатов расчета	

использование	248	спецификация материалов в	
Окружности		редакторе схем	181
в САМtastic	766	сравнение списков соединений ..	183
оптимизация использования термов		Отчеты	
произведения	407	в САМtastic	758
опции компилятора		ошибки	
проектирование ПЛИС	406	исправление ошибок	
ортогональное размещение		синхронизатора проекта	175
проводников на схемах	95	при моделировании схем	295
основная надпись		П	
добавление на схему информации о		панели инструментов	
разработчике	87	обзор	38
отладка		создание, пример	43
SimCode	317	Панелизация	
отменить		в САМtastic	779
в редакторе плат	546	Панель просмотра	
в редакторе схем	120	использование	10
относительное начало координат ..	497	Панель управления Project Workspace	
отображение		в САМtastic	725
видимые сетки	504	панорамирование	
линий связи	504	автоматическое в редакторе плат	516
маркера начала координат	509	в редакторе принципиальных схем	
номеров контактных площадок ..	509	91
отверстий контактных площадок и		ручное в редакторе плат	516
переходных отверстий	504	папки	
ошибки DRC	504	использование	11
отображение примитивов на платах		параметры	
.....	510	редактор принципиальных схем ..	82
отображение слоев	500	параметры объектов по умолчанию	510
отображение формы сигнала	249	параметры узла	247
отчеты		пары слоев для сверления отверстий	
Bill Of Materials в редакторе схем		581
.....	181	переключатели	
Board Information	680	управляемые напряжением	282
статус списка цепей	681	управляемые током	282
netlist status	681	перемещение	
Selected Pins	680	вершины полигона	539
в редакторе схем	181	объектов на платах	537
выделенные выводы	680	объектов на схемах	112
иерархия проекта	182	примитивов на платах	538
измерение расстояний на плате ..	540	разбиение проводника	539
информация о плате	680	точка захвата компонента	505
о библиотеке элементов схем	143	точка удержания объекта	83
о проверке компонентов схем	143	Перемещение	
об элементе схемы	143	в САМtastic	771

перемычки		копирование чертежа платы в буфер обмена Windows	690
проектирование ПЛИС.....	394	описание	578
перенос компонентов на другой слой	623	присвоение обозначение	710
перерисовка слоев в редакторе плат	509	создание новой.....	512
перетаскивание		сохранения в старых форматах ..	512
объектов на платах	537	печать	
объектов на схемах	113	в САМtastic	744
ортогональное на схеме.....	82	в редактре схем	177
переход		границ листов принципиальных схем.....	85
в абсолютное начало координат ..	517	изменение принтера при печати чертежа платы	689
в заданную точку схемы по координатам	93	настройка печати чертежей печатных плат	688
в заданную точку чертежа платы.....	517	настройка принтера при печати чертежа платы	690
в начало координат на схеме	93	чертежей печатных плат	686
в новую точку чертежа платы.....	517	печать чертежей плат	693
в редакторе принципиальных схем	93	платы	
к выделенному объекту	518	границы размещения и трассировки	579
к компоненту на плате.....	517	добавление слоев	579
к контактной площадке компонента	518	параметры слоев	581
к маркеру ошибки на плате.....	518	пары слоев для сверления отверстий.....	581
к текстовой строке на плате.....	518	присвоение обозначений.....	710
к цепи на плате.....	517	расположение проводящих и изолирующих слоев.....	581
переходные отверстия		создание с помощью мастера Board Wizard	582
глухие и скрытые	552	стек слоев	579
максимальное число	607	физические границы	578
обзор	552	плоский проект	149
отображение отверстий	504	площадь металлизации	
под SMD элементами	608	в САМtastic	758
размещение.....	553	по умолчанию	
редактирования	553	позиционное обозначение.....	141
соединение с внутренним слоем питания	656	сброс пользовательских настроек ..	40
стиль.....	601	шаблон листа принципиальной схемы	83
перечни		по умолчанию	
настройка	700	примитивы на платах	510
периодических импульсов, источники	260	поворот	
перьевые плоттеры	709	установка шага.....	506
печатные платы		Поворот	
вывод на печать.....	686	в САМtastic	772
загрузка старых проектов.....	512		

поворот компонентов на платах	623	помадочные места	
повторная заливка полигонов	562	создание с помощью мастера	575
повторная трассировка		порт	
существующих проводников	653	обзор	123
повторное присвоение обозначений		порт питания	
.....	710	обзор	123
подтверждение глобального		порты	76
редактирования	506	связанность	78
подчиненные листы		создание связанности	148
добавление	147	порты питания	76
подчиненный лист	151	создание связанности	148
обзор	145, 146	порядок прорисовки слоев	509
позиционное обозначение		посадочные места	
в редакторе схем	133	обновление	576
позиционные метки		создание вручную	575
использование	93	права доступа	
переход к метке	93	база данных проекта	24
позиционные обозначения		правила проектирования	
присвоение в редакторе схем	165	Acute Angle Constraint	602
поиск		clearance constraint	597
компонента в библиотеках	131	Component Clearance Constraint ..	608
поиск и замена		Component Orientation Rule	609
текста на схемах	109	Daisy Chain Stub Length	606
поиск нарушений правил		high speed	606
проектирования	675	hole size constraint	603
поиск разорванных цепей	675	Impedance Constraint	611
полевой транзистор (MESFET)		Layer Pairs	603
моделирование	282	Length Constraint	606
полевой траннзистор (JFET)		manufacturing	602
моделирование	280	Matched Net Lengths	607
полигоны		Maximum Via Count	607
соединbnt с цепью	561	Minimum Annular Ring	603
в редакторе печатных плат	559	Nets to Ignore	609
изменение границ	562	Other	614
на сигнальных слоях	659	Overshoot — Falling Edge	611
перемещение вершин	539	Overshoot — Rising Edge	611
повторная заливка	562	Parallel Segment Constraint	608
прокладывание проводников внутри		Paste-Mask Expansion Rule	604
.....	653	Permitted Layers Rule	610
размещение	560	placement	608
соединение с контактными		Polygon Connect Style	604
площадками	562	Power Plane Clearance	604
стиль соединения выводов с		Power Plane Connect Style	605
полигоном	604	Room Definition	610
Полигоны		Routing	597
в САМtastic	766	Routing Corners Rule	598

Routing Layers Rule.....	598	максимальное число переходных отверстий.....	607
Routing Priority Rule.....	598	минимальное расстояние до изгиба	601
Routing Topology Rule.....	599	минимальный размер контактной площадки.....	603
Routing Via Style Rule.....	601	мониторинг.....	593
Routing Width Constraint.....	602	напряжение сигнала высокого уровня.....	613
Short Circuit Constraint.....	614	неприсоединенные выводы.....	615
Signal Base Value.....	611	неразведенные цепи.....	615
Signal Flight Time Falling Edge....	612	низкое значение уровня сигнала	611
Signal Flight Time Rising Edge....	612	обзор.....	583
Signal Stimulus.....	613	области действия.....	586
Signal Top Value.....	613	области размещения.....	610
Slope — Falling Edge.....	613	ограничение длины проводника.	606
Slope — Rising Edge.....	613	ограничение на длину параллельных сегментов.....	608
SMD Neck-Down Constraint.....	601	ограничение на длину шлейфа...	606
SMD to Corner Constraint.....	601	ограничение на размер острых углов.....	602
SMD to Plane Constraint.....	602	ограничение ширины проводников	602
Solder-Mask Expansion Rule.....	605	определение области действия...	586
Testpoint Style.....	605	ориентация компонентов.....	609
Testpoint Usage.....	606	отрицательный выброс на заднем фронте импульса.....	611
Un-Connected Pin Constraint.....	615	отрицательный выброс на переднем фронте импульса.....	614
Undershoot — Falling Edge.....	614	пары слоев.....	603
Undershoot — Rising Edge.....	614	переходные отверстия под SMD элементами.....	608
Un-Routed Nets Constraint.....	615	положительный выброс на заднем фронте импульса.....	614
Via Under SMD Constraint.....	608	положительный выброс на переднем фронте импульса....	611
анализ целостности сигналов.....	610	применение.....	592
в САМtastic.....	756	примеры использования.....	615
входные сигналы.....	613	приоритет трассировки.....	598
выключение.....	594	приоритеты областей действия..	591
диаметр отверстий.....	603	просмотр прилагаемых к объекту	594
длины согласованных цепей.....	607	размер окна в трафарете Paste-Mask	604
добавление.....	584	размер окна в трафарете Solder- Mask.....	605
дублирование.....	592		
задержка заднего фронта.....	612		
задержка переднего фронта.....	612		
зазоры.....	597		
зазоры на слоях питания.....	604		
игнорирование цепей.....	609		
импорт и экспорт.....	594		
использование контрольных точек	606		
короткозамкнутые цепи.....	614		
крутизна заднего фронта импульса	613		
крутизна переднего фронта импульса.....	613		

разрешение конфликтов	592	прямоугольные области заливки	554
разрешенные слои	610	редактора печатных плат	546
расстояние до переходного отверстия	602	текстовые строки на платах	557
расстояние между компонентами	608	установки по умолчанию для плат	510
скачок ширины проводника	601	установки по умолчанию для схем	84
слои трассировки	598	примитивы по умолчанию на схемах	84
составная область действия	590	принципиальные схемы	
стиль контрольных точек	605	библиотеки ПЛИС	367
стиль переходных отверстий	601	верификация	158
стиль соединения выводов с полигоном	604	обзор	74
стиль соединения выводов со слоем питания	605	подготовка к разработке платы ..	165
топология трассировки	599	проектирование ПЛИС	367
требования к импедансам	611	проблемы	
углы изгиба проводников	598	анализ переходных процессов ..	298
унарные и бинарные	589	сходимости в SPICE	295
устранение нарушений	675	проверка правил проектирования обзор	672
предварительный просмотр схемы перед печатью	180	проверка схем	158
предохранители		проводник	
моделирование	284	обзор	122
предупреждения	299	разбиение на схемах	100
во время моделирования	306	проводники	
преобразование		автоматическое размещение на схемах	95
проекты старых версий Protel	30	автосоединение на схемах	96
примеры		изгибы	549
виртуальное устройство	486	изменение	550
двухрядный счетчик	472	разбиение в редакторе плат	539
десятичный счетчик	475	размещение с предсказанием	650
дешифратор ЖКИ	482	режимы размещения	549
использования правил проектирования	615	режимы размещения на схемах ..	95
контроллер турникета	474	связанность	77
проектов на ПЛИС	465	удаление излома на схемах	100
простые логические элементы	465	программа моделирования	
четырёхрядный счетчик	484	возможности	211
примитивы		проект	
в редакторе схем	121	база данных	14
дуги на платах	555	импорт документов	21
контактные площадки pad	550	права доступа к базе данных проекта	24
линии	546	работа с проектами старых версий Protel	30
перемещение в редакторе плат ..	538	редактирование	19
переходные отверстия	552		

связывание документов.....	22
создание иерархической структуры	144
создание нового документа.....	17
создание нового проекта.....	16
экспорт документа.....	22
проектирование ПЛИС	
библиотеки элементов схем.....	367
выбор целевой микросхемы.....	403
выходные форматы.....	410
загружаемые файлы.....	411
компилятор.....	364
компиляция.....	403
конечные автоматы.....	400
многолистовые проекты.....	387
моделирование.....	413
на основе схемы.....	367
настройка и запуск компилятора	366
настройки для компиляции.....	403
обзор.....	364
основные возможности.....	361
поиск устройств.....	405
примеры.....	443, 465
синтаксис языка CUPL.....	396
стыковка выводов.....	387
элементы схем.....	369
язык CUPL HDL.....	389
просмотр	
в SAMtastic.....	753
принципиальных схем.....	92
просмотр платы с помощью панели управления.....	516
Просмотр проекта	
в SAMtastic.....	729
просмотр чертежа платы.....	514
простая иерархия.....	151
процедуры	
запуск.....	46
обзор.....	45
параметры.....	47
прямое аннотирование.....	169
Прямоугольники	
в SAMtastic.....	765
Р	
Работа с буфером обмена	
в SAMtastic.....	746
Работа с мышью	
в SAMtastic.....	728
Работа с проектами	
в SAMtastic.....	732
разбиение большой схемы для печати на нескольких листах.....	180
разбиение проводников на схемах.....	100
разделенные слои питания.....	656
размеры	
листов принципиальных схем.....	84
на чертежах печатных плат.....	563
размещение.....	563
редактирование.....	563
Размеры	
в SAMtastic.....	767
размещение	
автоматическое с помощью Cluster Placer.....	631
автоматическое с помощью Global Placer.....	632
автоматическое на платах.....	630
интерактивное на платах.....	628
компонентов на платах вручную.....	622
контактных площадок.....	551
обмен информацией между схемой и платой.....	625
опции.....	622
размещение компонентов на платах	622
разработка принципиальных схем	
основы.....	74
разрешения	
определение.....	26
растровые фотопроттеры.....	704
расчет комплексных сопротивлений	
настройка и запуск.....	241
расчет рабочей точки	
настройка и запуск.....	227
регистрация.....	5
редактирование	
атрибутов цепей на платах.....	642
во время размещения на плате ...	519
горячие клавиши редактора плат.....	541

горячие клавиши редактора	
принципиальных схем.....	116
графическое в редакторе плат	520
графическое объектов на схемах	100
документов	19
объектов во время размещения на	
схеме	98
объектов на платах	519
объектов на схемах	97
размещенных объектов на схемах	98
Редактор библиотек	
элементы принципиальных схем	135
редактор библиотек топологических	
посадочных мест	573
редактор библиотек элементов	
принципиальных схем	135
редактор печатных плат	
САМ Manager	693
границы размещения и трассировки	
.....	579
добавление слоев	579
компоненты и библиотеки	565
координаты.....	497
обновление платы после изменения	
схемы	169
основные возможности	489
параметры слоев	581
пары слоев для сверления отверстий	581
просмотр трехмерного вида.....	682
работа в нем.....	514
расположение проводящих и	
изолирующих слоев.....	581
стек слоев	579
топологические директивы .	168, 173
Редактор печатных плат	
информация из редактора схем	80
редактор принципиальных схем	
обзор	70
присвоение позиционных	
обозначений.....	165
прямое аннотирование	169
работа в нем.....	91
характеристики	70
режимы размещения проводников.	549
резисторы	
моделирование	276
реле	
моделирование.....	285
ресурсы	
обзор	38
редактирование, пример	43
установки по умолчанию	40
Рисование	
в САМtastic	764
рука в редакторе печатных плат.....	543
С	
сброс выделения	
объектов на схемах.....	101
перед глобальным редактированием	
.....	109
сброс выделения в редакторе плат.	522
сброс текущего начала координат .	497
связанность	
логическая.....	77
между листами.....	147
на печатных платах	636
обзор	76
отображение.....	637
правила определения.....	77
физическая	77
связанные катушки индуктивности,	
моделирование.....	246
связывание	
документа с базой данных проекта	
.....	22
связь	
элемента с подчиненным листом	133
секции	
атрибуты в редакторе схем	132
компонентов в редакторе схем ...	133
ориентация	134
размещение на схемах.....	131
текстовые поля.....	134
топологическое посадочное место	
по умолчанию	141
секции компонентов	
обзор	75
серверы	
запуск и останов	50
инсталляция	49
работа с серверами	48

удаление.....	50	сложная иерархия.....	153
сессии		слои	
мониторинг открытых документов		keep out	504
.....	29	multi	504
сетка		активизация.....	500
в САМtastic.....	751	внутренние	502
видимая в редакторе схем.....	86	границ размещения и трассировки	
размещения в редакторе схем.....	86	504
шаг штриховки полигона	561	маски для нанесения паяльной	
электрическая в редакторе плат ..	643	пасты.....	503
электрическая в редакторе схем...	86	маски для пайки волной.....	503
электрическая, использование.....	96	механические	502
сетка размещения		отображение.....	500
редактор принципиальных схем...	86	перемещение между слоями.....	501
сетки		питания и заземления.....	502
в редакторе печатных плат.....	498	порядок прорисовки	509
видимые в редакторе плат.....	499	прозрачные.....	509
координатная.....	498	сверления Drill Guide	503
при трассировке	646	сверления отверстий Drill Drawing	
размещения компонентов	498	503
электрическая в редакторе плат ..	498	текущий	501
символ листа		только один слой	509
обзор	123	шелкографии	502
символы листа		Слои	
использование	146	в САМtastic	749
синтаксис		слои питания	
выделение цветом	59	зазоры на слоях питания	604
синтаксис языка		стиль соединения выводов со слоем	
SimCode	317	питания	605
синусоидальные источники	259	слой сверления Drill Guide.....	503
синхронизатор проекта		слой сверления отверстий Drill	
использование	169	Drawing.....	503
исправление ошибок.....	175	случайное значение	
предупреждения.....	169	моделирование ПЛИС.....	421
синхронизация		совместимость	
моделирование ПЛИС	422	работа с проектами старых версий	
система единиц		Protel	30
в редакторе печатных плат.....	497	Совместимость с пакетом SPICE ...	212
системные требования.....	4	Совмещение слоев	
системный шрифт в редакторе схем		в САМtastic	772
.....	128	соединения	
скрытые выводы		межлистовые	147
отображение на схемах	134	создание	
создание связанности	148	элемента принципиальной схемы	
Слияние слоев		139
в САМtastic.....	774		

создание нового SimCode устройства	308
сообщения	
ошибки и предупреждения	299
сообщения об ошибках	299
Сопряжение линий	
в SAMtastic	770
составная область действия правил проектирования	590
специальные строки	
в редакторе печатных плат	508
ввод содержимого на схемах	87
использование на схемах	87
отображение содержимого	83
специальные строки в редакторе плат	558
специальные строки на схемах	88
списки соединений	
генерация	717
загрузка	713
изменение и обновление	641
макросы синхронизатора проекта	714
обзор	195
ограничения формата	717
создание	195
формат Protel	198
формат Protel 2	198
форматы	196
формирование	195
экспорт	717
справочная система	6
среда проектирования	
изменение настроек	38
статистический анализ	
настройка и запуск	228
стек контактных площадок	551
стек слоев	509, 579
строка состояния	
координаты в редакторе печатных плат	497
Строка состояния	
в SAMtastic	727
строки	
отображение содержимого	83
специальные на схемах	88
схемы	
обновление измененных элементов	143
обновление по изменениям в плате	710
обновление по изменениям на плате	711
ошибки моделирования	295
сходимость	
отсутствие	296
T	
Таблица инструментов	
в SAMtastic	752
текст	
библиотечные поля компонентов	134
библиотечные поля на схемах	141
в SAMtastic	767
добавление на схему информации о разработчике	87
изменяемые текстовые поля	142
на чертежах печатных плат	557
отображение скрытых текстовых полей на схемах	134
поиск и замена текста на схеме ..	109
присвоение позиционных обозначений	165
расположение названий выводов ..	82
редактирование непосредственно на листе	98
текстовые поля компонентов	132, 134
текстовые строки	
переход к нужной	518
размещение на печатных платах ..	558
редактирование на платах	558
специальные на платах	558
текстовый редактор	58
выделение цветом синтаксиса	59
параметры документов	60
языки	58
тепловые барьеры	
на внутренних слоях питания	655
термы производства	
количество	412

ограничения.....	406	подготовка платы	646
технология Smart		приоритет	598
обзор	34	прокладка проводников с	
тип элемента		предсказанием.....	650
в редакторе схем	133	прокладывание проводников внутри	
ток потребления		полигона	653
моделирование	305	размещение компонентов в сетку	
переменная перекрытия	306	647
топологии цепей		расталкивание препятствий	652
обзор	637	ручная	643, 648
определение.....	638	скачок ширины проводника	601
топологическая директива		слои	598
обзор.....	124	стиль переходных отверстий.....	601
топологические директивы	80, 168	топология	599
топологическое посадочное место		углы изгиба проводников	598
задание в редакторе схем	132	трехмерный вид платы	682
по умолчанию	141		
точка привязки при копировании.....	83	у	
точка соединения		угловое направление на слое.....	666
обзор	122	удаление	
точки данных		объектов в редакторе плат	539
отображение	255	удаление излома проводника на схеме	
транзисторы		100
моделирование	279	удаление листа схемы из проекта ..	147
трансформаторы		уменьшение избыточности	
моделирование	286	логической схемы.....	409
трассировка		управление проектами	
бессеточная.....	646	обзор	144
блокирование ранее разведенных		управляемый напряжением генератор	
проводников	661	прямоугольных импульсов	270
горячие клавиши	649	управляемый напряжением генератор	
диагональная	666	треугольных импульсов	271
изменение параметров проводников		управляемый напряжением источник	
и переходных отверстий во время		напряжения	268
трассировки	648	управляемый напряжением источник	
карта плотности	647	тока	268
контроль связанности.....	644	управляемый напряжением	
минимальное расстояние до изгиба		синусоидальный источник.....	270
.....	601	управляемый током источник	
минимальное расстояние до		напряжения	269
переходного отверстия	602	управляемый током источник тока	269
настройка сеток.....	646	управляемый частотой источник	
обновление списка соединений ..	641	напряжения	269
ограничение ширины проводников		установка текущего начала координат	
.....	602	497
повторная.....	653	установки по умолчанию	

сброс текущего начала координат	497	переход к нужной	517
установки ресурсов по умолчанию		список загруженных	681
обзор	40	топологии	637
устройства		цифровое моделирование	
виртуальные	406	питание и заземление	300
выбор для проекта на ПЛИС	403	цифровые компоненты	
мнемоническое обозначения	405	моделирование	288
обозначение ПЛИС	405	цифровые схемы	
поиск ПЛИС	405	моделирование	307
учебник		Ч	
выделение объектов в окне	102	частные производные	305
выделение отдельных объектов на		частотно-модулированных сигналов,	
схемах	102	источники	266
глобальное редактирование плат	534	Ш	
поиск и замена текста на схемах	109	шаблон	
разбиение проводников на схемах		копирование в буфер обмена	83
.....	100	листа принципиальной схемы,	
размещение проводников	94	создание	88
создание пользовательского		текущий шаблон листа схемы	86
шаблона	88	шаблон листа	
удаление излома проводника на		создание	88
схеме	100	шаблоны	
Ф		задание шаблона, используемого по	
фаски		умолчанию	90
в САМtastic	769	изменение	90
физическая связанность		обзор	87
обзор	77	обновление	90
Флэши		удаление	90
в САМtastic	764	шаблоны листов принципиальных	
фокус		схем	
в редакторе печатных плат	520	установки по умолчанию	83
редактор принципиальных схем	98	шаг сетки	
форматы		в редакторе печатных плат	498
Gerber	705	шелкография	
Фрезерование		слои	502
в САМtastic	780	шина	
Фурье анализ		обеспечение связанности	122
настройка и запуск	237	обзор	122
Ц		шины	
цепи		связанность	78
внутренние слои питания	655	Шоттки транзистор	
длины согласованных цепей	607	моделирование	282
идентификация на платах	642	шрифты	

в редакторе принципиальных схем	127	в САМtastic	740
изменение системного шрифта.....	87	Экспорт списка соединений	
поворот	178	в САМtastic	743
системный шрифт редактора схем	128	Экспорт файлов	
текстовые поля		в САМtastic	740
компонентовcomponent text	132	Экспорт файлов NC Mill/Route	
Э		в САМtastic	743
экспоненциальных сигналов,		электрическая сетка	
источники	264	в редакторе печатных плат	498
экспорт		использование	96
документа из базы данных проекта	22	редактор принципиальных схем ..	86
Экспорт BMP файлов		электрические объекты	76
в САМtastic.....	744	элементы принципиальных схем ...	128
Экспорт DXF файлов		Эллипсы	
в САМtastic.....	741	в САМtastic	766
Экспорт Gerber файлов		Я	
в САМtastic.....	742	языки	
Экспорт NC Drill		редактирование.....	58
		синтаксис языка CUPЛ.....	396
		управления фотоплоттерами	705
		элементы языка CUPЛ.....	394

Содержание

ОТ АВТОРА	1
ГЛАВА 1	2
ИНСТАЛЛЯЦИЯ И НАЧАЛО РАБОТЫ	2
<i>Что такое система Protel 99 SE?</i>	2
<i>Инсталляция</i>	4
<i>Системные требования</i>	4
<i>Что входит в комплект поставки системы Protel 99 SE?</i>	4
<i>Инсталляция системы Protel 99 SE</i>	5
ГЛАВА 2	7
РАБОТА В СРЕДЕ ПРОЕКТИРОВАНИЯ DESIGN EXPLORER	7
<i>Среда проектирования Design Explorer</i>	7
<i>Как работать в среде Design Explorer?</i>	8
<i>Использование панели просмотра дерева проекта</i>	10
<i>Работа с окном проекта</i>	11
<i>Что такое база данных проекта?</i>	14
<i>Создание новой базы данных проекта</i>	16
<i>Выбор способа хранения документов</i>	16
<i>Защита базы данных проекта паролем</i>	17
<i>Создание нового документа внутри базы данных проекта</i>	17
<i>Редактирование документов</i>	19
<i>Импорт внешних документов в базу данных проекта</i>	21
<i>Добавление связей с внешними документами</i>	22
<i>Экспорт документов из базы данных проекта</i>	22
<i>Управление документами внутри базы данных проекта</i>	23
<i>Использование папок для построения иерархии</i>	23
<i>Перемещение, копирование и удаление документов</i>	24
<i>Переименование документа или папки</i>	24
<i>Настройка прав доступа при групповой разработке проекта</i>	24
<i>Определение команды разработчиков</i>	25
<i>Определение разрешений для команды разработчиков</i>	26

Мониторинг групповой разработки проекта	29
Блокирование документов	30
Работа с ранее созданными проектами системы Protel	30
Преобразование существующего проекта в базу данных проекта	31
Связывание с внешними документами	32
Работа с существующими библиотеками элементов	32
Синхронизация ранее созданных проектов схем и печатных плат	32
Технологии, используемые внутри системы Protel 99 SE	34
Технология SmartTool	34
Технология SmartDoc	37
Технология SmartTeam	37
Изменение настроек среды проектирования Design Explorer	38
Ресурсы — панели инструментов, меню команд и горячие клавиши	38
Понятие процедуры	45
Работа с серверами	48
Создание пользовательских серверов управления ресурсами	51
Текстовый редактор системы Protel 99 SE	58
Языки программирования и описания	58
Функция выделения цветом	59
Параметры документов	60
Работа с макросами	60
Пример программирования на языке Client Basic	60
ГЛАВА 3	70
РАЗРАБОТКА ПРИНЦИПИАЛЬНЫХ СХЕМ	70
Основные возможности редактора принципиальных схем	70
Редактор электрических принципиальных схем	70
Редактор библиотек элементов принципиальных схем	70
Возможности проектирования	70
Основы разработки принципиальных схем	74
Представление электрической схемы в компьютере	74
Модели элементов принципиальных схем	75
Связанность	76
Использование связности	77
Верификация проекта	79
Связь между редактором схем и редактором печатных плат	80
Настройки редактора принципиальных схем	82
Параметры редактора	82
Параметры листов принципиальных схем	84
Шаблоны листов принципиальных схем	87
Работа с редактором принципиальных схем	91

<i>Изменение масштаба просмотра листа</i>	91
<i>Перемещение в редакторе принципиальных схем</i>	91
<i>Размещение объектов на принципиальной схеме</i>	94
<i>Редактирование объектов на принципиальных схемах</i>	97
<i>Глобальное редактирование</i>	105
<i>Быстрое копирование объектов</i>	109
<i>Поиск и замена текста</i>	109
<i>Выравнивание объектов</i>	111
<i>Перемещение и перетаскивание</i>	112
<i>Горячие клавиши редактора принципиальных схем</i>	116
<i>Вложенные команды</i>	116
<i>Отмена обновления экрана</i>	116
<i>Горячие клавиши для мыши</i>	116
<i>Горячие клавиши для клавиатуры</i>	117
<i>Наиболее часто используемые горячие клавиши</i>	119
<i>Операции Undo и Redo</i>	120
<i>Объекты редактора принципиальных схем</i>	121
<i>Электрические примитивы</i>	122
<i>Графические (не электрические) примитивы</i>	124
<i>Шрифты</i>	127
<i>Библиотеки элементов принципиальных схем</i>	128
<i>Что такое библиотека элементов принципиальных схем?</i>	128
<i>Что такое компонент и секция?</i>	129
<i>Где находятся библиотеки компонентов принципиальных схем?</i>	129
<i>Доступ к компонентам, необходимым для конструирования</i>	130
<i>Размещение секций компонентов на листе принципиальной схемы</i>	131
<i>Атрибуты элементов принципиальных схем</i>	132
<i>Редактор библиотек элементов принципиальных схем</i>	135
<i>Управление многолистовыми и иерархическими проектами</i>	144
<i>Общие положения</i>	144
<i>Управление многолистовыми проектами</i>	145
<i>Структура многолистовых принципиальных схем</i>	145
<i>Главные и подчиненные листы</i>	146
<i>Как создается связанность в многолистовом проекте?</i>	147
<i>Различные методы построения многолиствого проекта</i>	149
<i>Работа с иерархическим проектом</i>	156
<i>Верификация принципиальных схем</i>	158
<i>Определение точек, которые не будут восприниматься как ошибки</i>	159
<i>Опции программы верификации</i>	159
<i>Настройка области действия идентификаторов цепей</i>	161
<i>Настройка матрицы правил электрических соединений</i>	161
<i>Исправление ошибок</i>	163
<i>Подготовка к проектированию печатной платы</i>	165
<i>Обозначение элементов</i>	165
<i>Проверка наличия топологических посадочных мест элементов</i>	167

Проверка правил электрических соединений	167
Включение в схему топологических директив	168
Подготовка к проектированию печатной платы	168
Передача информации о схеме в редактор печатных плат	169
Передача информации о схеме	169
Как синхронизатор определяет, какой чертеж печатной платы использовать?	173
Как размещаются компоненты на чертеже печатной платы?	173
Передача топологической информации	173
Передача изменений в принципиальной схеме в редактор печатных плат	174
Как синхронизатор сопоставляет компоненты на принципиальной схеме и на печатной плате?	174
Как синхронизатор передает проектную информацию	175
Печать принципиальных схем	177
Обзор	177
Вывод на принтер или плоттер	177
Создание отчетов	181
Спецификация материалов (Bill of Materials)	181
Перекрестные ссылки (Cross Reference)	182
Иерархия проекта (Design Hierarchy)	182
Сравнение списков соединений (Netlist Compare)	183
Связь с внешними базами данных	183
Горячая связь с базой данных	183
Импорт и экспорт баз данных	186
Обмен данными с другими системами проектирования	190
Преобразование информации из OrCAD	191
Обмен информацией с системой P-CAD 2000	193
Интерфейс с механическими САПР	195
Создание списков соединений	195
Некоторые рекомендации по разработке схем по ГОСТ	201
Настройки сеток	201
Разработка библиотечных элементов	203
Прорисовка портов питания	207
Особенности разработки иерархических проектов	209
ГЛАВА 4	211
МОДЕЛИРОВАНИЕ СМЕШАННЫХ СИГНАЛОВ	211
Моделирование смешанных сигналов — основные возможности	211
Введение в моделирование	214
Примеры схем	214
Создание схемы	214

Ввод источников сигналов	215
Установка точек контроля сигналов	216
Установки параметров анализа	216
Запуск процесса моделирования	216
Просмотр результатов моделирования	217
Подготовка и проведение моделирования	218
Сбор, отображение и сохранение результатов моделирования	218
Анализ переходных процессов (Transient Analysis)	220
Частотный анализ схемы в режиме малого сигнала (AC Sweep)	223
Анализ схем при изменяющемся постоянном напряжении (DC Sweep)	225
Расчет рабочей точки по постоянному току (DC Operating Point)	227
Статистический анализ методом Monte Carlo	228
Автоматическое изменение параметров элементов (Parameter Sweep)	233
Режим изменения температуры (Temperature Sweep)	236
Анализ Фурье (Fourier Analysis)	237
Анализ передаточных функций (Transfer Function Analysis)	239
Анализ шумов (Noise Analysis)	240
Расчет комплексных сопротивлений (Impedance Plot Analysis)	241
Математические действия с рассчитанными сигналами	243
Включение дополнительной SPICE информации в список соединений	246
Создание списка соединений и запуск процесса моделирования	246
Установка начальных условий	247
Окно просмотра результатов расчета	248
Отображение формы сигнала	249
Масштабирование кривых	250
Просмотр сигналов на отдельных графиках	252
Отображение нескольких сигналов на одном графике	253
Одновременное отображение кривых, имеющих разный масштаб по оси Y	254
Изменение формата отображения результатов расчета	254
Отображение точек данных моделирования	255
Распознавание графиков при монохромной печати	255
Использование измерительных маркеров	256
Способ отображения результатов нескольких проходов моделирования	256
Источники напряжения и тока	257
Размещение компонентов источников	257
Источник постоянного напряжения или тока	258
Источник синусоидального сигнала	258
Источник синусоидального сигнала	259
Источник периодических импульсов	260
Источник кусочно-линейного сигнала	262
Источник сигнала экспоненциальной формы	264
Источник частотно модулированного сигнала	266
Линейные зависимые источники	268

<i>Нелинейные зависимые источники</i>	272
<i>Компоненты и модели</i>	274
<i>Библиотеки символов компонентов</i>	274
<i>Модели и подсхемы</i>	276
<i>Описания устройств</i>	276
<i>Создание собственных компонентов и подготовка их к моделированию</i>	289
<i>Обработка ошибок, возникающих при моделировании</i>	295
<i>Ошибки при генерации списка цепей</i>	295
<i>Сходимость программы SPICE</i>	295
<i>Стратегия решения проблем сходимости</i>	296
<i>Устранение ошибок, возникающих при анализе схемы по постоянному току</i>	298
<i>Устранение ошибок, возникающих при анализе переходных процессов</i>	298
<i>Предупреждения и сообщения об ошибках</i>	299
<i>Переменные и параметры модуля SPICE</i>	299
<i>Моделирование цифровых схем</i>	307
<i>Модели цифровых компонентов</i>	307
<i>Источники питания цифровых устройств и общий провод</i>	307
<i>Создание нового SimCode устройства</i>	308
<i>Определение языка SimCode</i>	314
<i>Синтаксис языка SimCode</i>	317
ГЛАВА 5	361
ПРОЕКТИРОВАНИЕ УСТРОЙСТВ НА БАЗЕ ПЛИС	361
<i>Основные особенности проектирования устройств на базе ПЛИС</i>	361
<i>Описание проектов на базе ПЛИС с помощью принципиальных схем</i>	362
<i>Описание проектов на базе ПЛИС с помощью языка высокого уровня</i>	362
<i>Трассировщик-компилятор ПЛИС</i>	362
<i>Программа моделирования устройств на базе ПЛИС</i>	362
<i>Просмотр временных диаграмм, полученных в результате моделирования</i>	363
<i>Язык высокого уровня CUPL</i>	363
<i>Поддерживаемые устройства</i>	363
<i>Обзор процесса проектирования устройств на базе ПЛИС</i>	364
<i>Ввод проекта</i>	364
<i>Компиляция проекта</i>	364
<i>Моделирование проекта</i>	366
<i>Где настраивается и запускается компилятор ПЛИС</i>	366
<i>Проектирование устройств на ПЛИС на основе принципиальных схем</i>	367
<i>Использование библиотеки символов</i>	367
<i>Разработка схемы внутренней логики</i>	386

Создание многолиствого проекта	387
Стыковка внутренней логической схемы с выводами компонента	387
Компиляция проекта ПЛИС на основе принципиальной схемы	388
Описание проекта устройства на ПЛИС на языке высокого уровня CUPL	
HDL	389
Обзор синтаксиса языка CUPL	390
Директивы исходного файла	391
Элементы языка CUPL	394
Синтаксис языка	396
Компиляция проекта на ПЛИС	403
Выбор целевой микросхемы	403
Опции компилятора	406
Форматы выходных файлов	410
Моделирование проекта на базе ПЛИС	413
Входные данные для программы моделирования	413
Результаты моделирования	413
Создание исходного файла тестовых спецификаций	414
Заголовок	414
Определение микросхемы	415
Комментарии	415
Операторы	415
Объявление переменных (VAR)	425
Директивы программы моделирования	425
Расширенный синтаксис	428
Операторы циклов	431
Виртуальное моделирование	438
Моделирование неисправностей	439
Просмотр смоделированных временных диаграмм сигналов	439
Пример сессии разработки проекта на базе ПЛИС	443
Этапы разработки	443
Шаг 1 — Проверка технического задания	444
Шаг 2 — Создание исходного кода на языке CUPL	446
Шаг 3 — Составление выражений	446
Шаг 4 — Выбор целевого устройства	448
Шаг 5 — Распределение выводов	450
Шаг 6 — Компиляция исходного файла	452
Шаг 7 — Создание проекта на основе принципиальной схемы	453
Шаг 8 — создание файла тестовых векторов для программы моделирования	457
Шаг 9 — моделирование устройства	461
Шаг 10 — просмотр сигналов после моделирования	464
Резюме	464
Примеры проектов на основе ПЛИС	465
Пример 1 — Простые логические элементы	465
Пример 2 — Двухразрядный счетчик	472

Пример 3 — Проект простого конечного автомата	474
Пример 4 — Двоично-десятичный реверсивный счетчик	475
Пример 5 — Дешифратор семисегментного жидкокристаллического индикатора	482
Пример 6 — Четырехразрядный счетчик с возможностью загрузки и сброса	484
ГЛАВА 6	489
ПРОЕКТИРОВАНИЕ ПЕЧАТНЫХ ПЛАТ	489
Редактор печатных плат — основные возможности	489
Редактор чертежей печатных плат	490
Редактор библиотек топологических посадочных мест	490
Бессеточный автоотрашиватель	490
Возможности редактора печатных плат	491
Настройки редактора чертежей печатных плат	497
Система координат	497
Точность прорисовки объектов	497
Переключение между дюймовой и метрической системой измерения	497
Система сеток	498
Работа со слоями	500
Настройка редактора печатных плат	504
Создание, открытие и сохранение чертежей печатных плат	512
Создание нового чертежа печатной платы	512
Сохранение копии чертежа под новым именем	512
Импорт и экспорт чертежей плат в формате ранних версий системы Protel	512
Работа в редакторе печатных плат	514
Изменение обзора рабочего пространства	514
Перемещение по чертежу печатной платы	515
Редактирование объектов на чертежах печатных плат	519
Комплексное выделение	522
Глобальное редактирование	532
Перемещение и перетаскивание	537
Непосредственное удаление объектов	539
Измерение расстояний на чертеже печатной платы	540
Горячие клавиши редактора печатных плат	541
Вложенные функции редактирования	541
Отмена процесса перерисовки экрана	541
Горячие клавиши для мыши	542
Инструмент Slider Hand (рука)	543
Горячие клавиши для клавиатуры	543
Специальные горячие клавиши, функционирование которых зависит от текущего режима работы	545

Размещение компонентов	545
Операции Undo и Redo	546
Объекты редактора печатных плат	546
Графические примитивы	546
Групповые объекты	559
Библиотеки топологических посадочных мест	565
Чем компонент отличается от топологического посадочного места?	565
Где находятся библиотеки топологических посадочных мест?	565
Доступ к посадочным местам, необходимым для конструирования	565
Поиск и размещение компонентов	567
Атрибуты компонентов печатных плат	568
Изменение назначенного топологического посадочного места	571
Модификация отдельных компонентов на печатной плате	571
Включение проводников в топологическое посадочное место	572
Разгруппировка компонента	572
Копирование компонентов с платы в библиотеку	572
Редактор библиотек топологических посадочных мест	573
Создание библиотеки проекта	577
Описание печатной платы	578
Задание физических границ платы	578
Определение границ размещения и трассировки	579
Задание стека слоев	579
Задание пар слоев для сверления отверстий	581
Мастер создания печатных плат Board Wizard	582
Передача данных из редактора принципиальных схем	583
Определение требований к проекту печатной платы	583
Что такое правило проектирования?	584
Определение правил проектирования	584
Применение правил проектирования	592
Работа с правилами проектирования	593
Классы объектов	595
Правила проектирования, учитываемые при трассировке (Routing)	597
Правила проектирования, учитываемые при производстве (Manufacturing)	602
Правила проектирования, задаваемые для высокоскоростных схем (High Speed)	606
Правила проектирования, используемые при размещении компонентов (Placement)	608
Правила проектирования, учитывающие анализ целостности сигналов (Signal Integrity)	610
Прочие правила проектирования (Other)	614
Примеры использования правил проектирования	615
Техника размещения компонентов и инструменты для этих целей	622
Основные опции размещения	622

Перемещение компонентов	623
Обмен информацией между принципиальной схемой и печатной платой	625
Использование групп компонентов	626
Работа с областями размещения	627
Интерактивное размещение компонентов	628
Автоматическое размещение компонентов	630
Что такое связанность и топология	636
Каким образом передается информация о компонентах и их связях	636
Как отображается информация о связях	637
Топологии цепей	637
Способы изменения топологии цепи	638
Пользовательские маршруты From-To	638
Отображение связей между выводами	640
Работа со списками соединений	641
Изменение атрибутов цепи	642
Идентификация цепей	642
Ручная трассировка печатной платы	643
Как редактор печатных плат работает со связями во время трассировки	644
Подготовка к трассировке	646
Трассировка печатной платы вручную	648
Режимы интерактивной трассировки	652
Повторная трассировка платы	653
Внутренние слои питания и заземления	655
Создание областей металлизации на сигнальном слое	659
Автотрассировка печатных плат	661
Настройка программы автотрассировки	661
Опции программы автоматической трассировки	664
Программа автоматической трассировки пакета Protel 99 SE — взгляд изнутри	665
Добавление контрольных точек и каплевидных контактных площадок	667
Что такое контрольная точка?	667
Настройка правил проектирования для контрольных точек	668
Подбор существующих объектов для использования в качестве контрольных точек	669
Добавление контрольных точек во время автотрассировки	670
Отчет о местоположении контрольных точек	670
Отчет о цепях, к которым контрольные точки добавлены не были	670
Удаление всех контрольных точек с платы	671
Создание контактных площадок и переходных отверстий каплевидной формы	671
Верификация проекта печатной платы	672
Что представляет собой модуль Design Rule Checker?	672
Интерактивная проверка правил проектирования	673

<i>Настройка пакетного режима проверки правил проектирования</i>	673
<i>Выполнение проверки в пакетном режиме</i>	674
<i>Отчет программы проверки</i>	674
<i>Устранение нарушений правил проектирования</i>	675
<i>Анализ целостности сигналов</i>	676
<i>Создание отчетов</i>	680
<i>Просмотр трехмерного вида печатной платы</i>	682
<i>Печать чертежа печатной платы средствами Windows</i>	686
<i>Настройки предварительного просмотра распечатки</i>	687
<i>Что такое распечатка (Printout)?</i>	687
<i>Изменение комплекта распечаток</i>	688
<i>Выбор принтера для печати</i>	689
<i>Настройка ориентации листа бумаги, масштаба и других параметров принтера</i>	690
<i>Печать чертежа</i>	690
<i>Копирование содержимого окна предварительного просмотра для работы с другими приложениями</i>	690
<i>Экспорт распечаток в виде метафайлов (WMF)</i>	691
<i>Цвета, шрифты и другие установки</i>	691
<i>Генерация выходных файлов для производства</i>	693
<i>Взаимодействие с производителями печатных плат</i>	693
<i>Генерация выходных файлов с помощью модуля CAM Manager</i>	693
<i>Создание нового САМ документа</i>	694
<i>Добавление настроек выходных файлов в САМ документ</i>	694
<i>Конфигурация опций процесса генерации выходных файлов</i>	696
<i>Генерация выходных файлов</i>	697
<i>Генерация выходных файлов в формате Gerber</i>	697
<i>Генерация файлов для сверления отверстий (NC Drill)</i>	700
<i>Генерация перечней используемых материалов (Bill of Materials)</i>	700
<i>Генерация файлов для оборудования размещения (Pick and Place)</i>	701
<i>Генерация отчета о контрольных точках (Testpoint Report)</i>	702
<i>Генерация отчета о проверке правил проектирования (Design Rule Check)</i>	703
<i>Перенос САМ документа в другой проект</i>	703
<i>Какие бывают фотошаблоны?</i>	704
<i>Передача изменений в проекте на этап разработки принципиальной схемы</i>	710
<i>Повторное присваивание обозначений компонентов</i>	710
<i>Обновление принципиальной схемы, исходя из изменений в печатной плате</i>	711
<i>Интерфейс с другими системами проектирования</i>	712
<i>Загрузка списка соединений</i>	713
<i>Экспорт списков соединений</i>	717
<i>Модули импорта и экспорта файлов</i>	718

ГЛАВА 7	720
ПРОГРАММА САМТАSTIC! 2000 DESIGNERS' EDITION	720
Общие сведения	720
Инсталляция программы	722
Основные понятия	723
Панель управления <i>Project Workspace</i>	725
Строка состояния	727
Работа с мышью	728
Горячие клавиши	729
Просмотр проекта	729
Выделение объектов	731
Работа с проектами	732
Импорт файлов	733
Импорт <i>NC Drill</i> файлов	734
Импорт <i>DXF</i> и <i>DWG</i> файлов	734
Импорт <i>Gerber</i> файлов	737
Импорт файлов в формате <i>HPGL/HPGL2</i>	739
Импорт файлов в формате <i>NC Mill/Route</i>	739
Импорт файлов апертур	740
Экспорт файлов	740
Экспорт файлов <i>NC Drill</i>	740
Экспорт <i>DXF</i> файлов	741
Экспорт <i>Gerber</i> файлов	742
Экспорт файлов в формате <i>NC Mill/Route</i>	743
Экспорт списка соединений	743
Экспорт <i>BMP</i> файлов	744
Вывод файлов на печать	744
Работа с буфером обмена	746
Работа в программе <i>CAMtastic</i>	746
Апертуры	746
Слои	749
Режимы захвата и рисования	751
Таблица инструментов	752
Режимы просмотра файлов	753
Инструменты верификации	755
Генерация отчетов	758
Инструменты редактирования	764

ЛИТЕРАТУРА	786
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	789
СОДЕРЖАНИЕ	817