

Система проектирования Altium Designer 6

Статья открывает цикл публикаций, представляющих собой краткие наброски последовательности действий при создании схемных проектов размещения элементов на плате и трассировки, подготовке выходных файлов в системе проектирования Altium Designer 6. Даны некоторые рекомендации по ведению проектов — для автоматизации подготовки сопровождающих проект документов. Для подробного изучения всех возможностей Altium Designer 6 приведены ссылки на конкретные разделы встроенных HELP или других оригинальных файлов, поставляемых с приложением.

Владимир Пранович

Pranovich@bsu.by

Создание и настройка проектов

Настройка панелей

Панелей в пакете Altium Designer 6 множество (GU0112 Welcome to the Altium Designer Environment; TR0104 Altium Designer Panels Reference). Все панели динамические и меняют свое наполнение в зависимости от активного документа, освобождая максимум пространства для основного окна (рис. 1). Более того, многие из них доступны только для своего класса активного документа. Настройка панелей легко производится на основе HELP, и более того, будет постоянно меняться в зависимости от вашего желания в процессе изучения и работы. В дальнейшем будут иногда даваться только рекомендации по настройкам панелей.

Системное меню и кнопки Shortcuts имеют классическое расположение сверху. Там же находится и окно навигатора документов. Строка сообщений со встроенной панелью вызова вспомогательных панелей для активного документа внизу, а в верхней части основного поля располагается строка с именами открытых документов для быстрого выбора активного документа.

Вспомогательные панели могут быть на всех четырех сторонах основного меню и иметь «выплывающий» характер или становиться прозрачными, не мешая редактировать активный документ. Более подробное описание панелей и функций будет показано в разделах, посвященных конкретным редакторам.

Выполнить какую-либо команду можно через многие меню, в том числе всплывающие, контекстные, и один из путей всегда легко находится через HELP. Здесь и далее будет всегда описан только один из них. Остальные пути пользователи легко найдут сами.

В приложении активно используются сочетания нажатия кнопок клавиатуры для быстрого выполнения наиболее часто используемых команд. Список приведен в документе GU0104 Shortcut Keys.pdf.

Создание Design Workspace

Design Workspace создает файл, где хранятся данные о файлах нескольких проектов, объединенных общей тематикой: например, несколько плат одного модуля. Наполнение Design Workspace проектами будет показано ниже на примерах. Это достаточно удобная форма объединения различных файлов, которые могут находиться в разных директориях, но, как правило, относятся к логически связанным проектам. Особенно удобна эта функция при поддержке большого количества проектов, так как с течением времени позволяет оперативно открыть все документы проекта. В Design Workspace могут быть включены не только файлы, созданные приложениями самого пакета, но и приложениями других программ, например Word, Excel, Access и т. п. Это позволяет создавать вспомогательные документы (перечни элементов, спецификации) и включать их в проект, легко редактируя с течением времени.

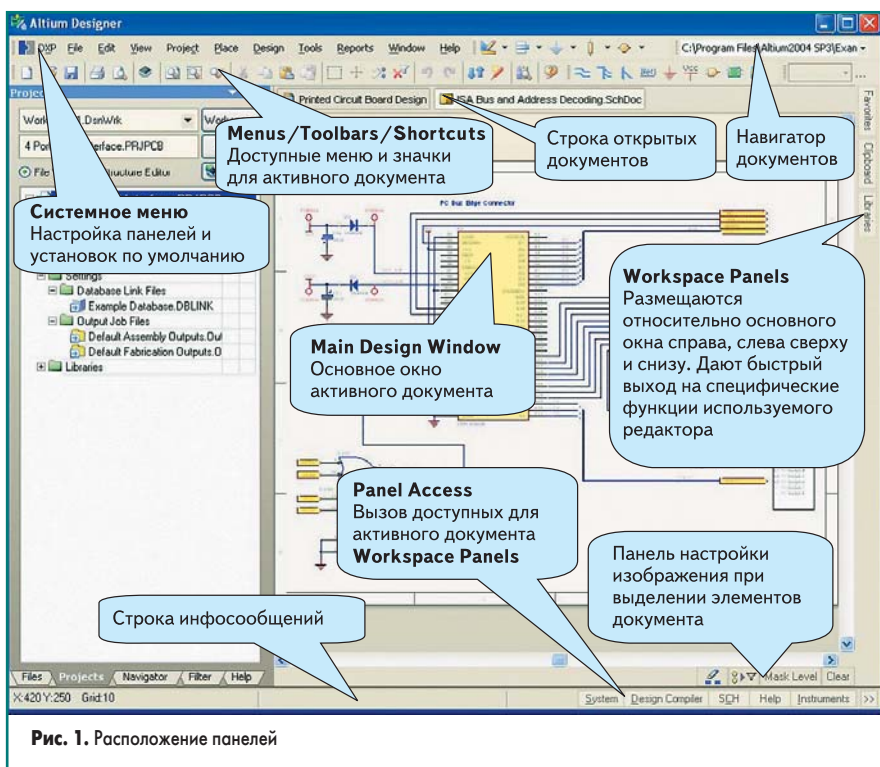


Рис. 1. Расположение панелей

Создание новой Design Workspace происходит при помощи команды **File/New/Design Workspace**. При этом создается файл **Workspace1.DsnWrk**, в котором будут описаны проекты, включенные в **Design Workspace**.

Хранить файл «*.DsnWrk» можно где угодно, но желательно организовать отдельную папку, в директории, определенной вами, например, «..\Altium\DsnWrk». Это даст возможность более просто и удобно держать ссылки на все типы проектов, которые вы будете вести в одной папке. После создания новой Design Workspace сразу же сохраните ее (**File/Save Design Workspace As**) в выбранную директорию, присвоив ей уникальное имя. Пусть в нашем случае это будет **Example**.

Создание нового проекта

Проект, как правило, включает в себя набор документов, объединенных каким-либо общим условием, например, все документы, созданные для проектирования, подготовки печатной платы, ее изготовления, чертежей, относящихся к ней, сборочных единиц и т. д.

Начнем с простых проектов, а именно:

- проекта, включающего типовые заготовки для создания схем;
- проекта, включающего созданные нами библиотеки;
- проектов, включающих созданные нами правила проектирования, правила формирования выходных данных, заготовок текстовых документов;
- непосредственно с нескольких простых проектов печатных плат.

Создание нового проекта (GU0112 Welcome to the Altium Designer Environment) **PCB Project** производится командой **File/New/Project/PCB Project**. Сразу сохраним его, например, в папку «..\Altium\Title» с именем **Title_GOST**. При первом сохранении возникает окно с предложением типа сохраняемого файла. В окне лучше выбрать галочку около надписи **Protel PCB**, хотя принципиального значения это не имеет.

Создание нового документа в проекте

Создадим (GU0112 Welcome to the Altium Designer Environment) новый документ в проекте, который в примере будет содержать рамку первого листа и основные надписи (по ГОСТ 2.104-68, далее сокращенно «**Title**») по команде **File/New/Schematic** и сохраним его (**File/Save As**) в папку «...\Altium\Title» с именем **A4V1** (лист A4, вертикальное расположение листа, **Title** для 1 листа). При сохранении следует выбрать тип **Advanced Schematic template**. Это позволит в дальнейшем использовать этот файл в качестве заготовки при создании схем. При создании непосредственно схемы следует сохранять ее как **Advanced Schematic binary** или **Advanced Schematic ascii**. В настройках **DXP/Preferences/Schematic/Graphical Editing** графического редактора схем снимите галочку около **Convert Special String**. Это необходимо для правильной расстановки положения надписей согласно ГОСТ. При редактировании самой схемы ее нужно будет установить обратно.

С помощью команды **Place Line** панели **Utilities** (AP0109 Schematic Editing Essentials)

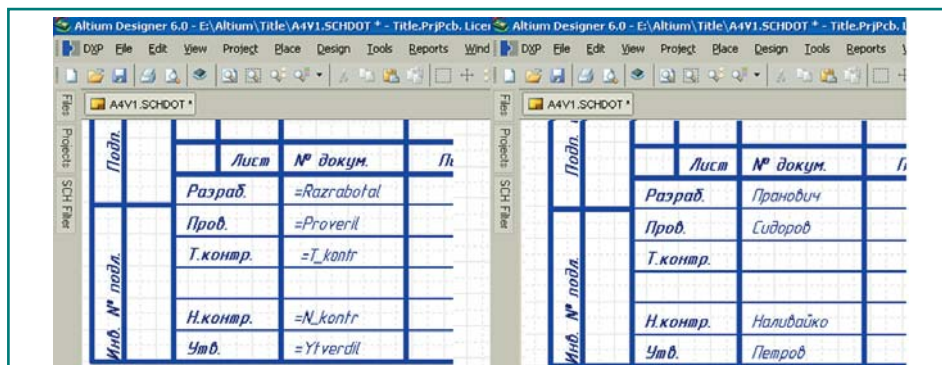


Рис. 2. Пример заполнения полей основной надписи и ее вид после установки галочки около **Convert Special String**

нарисуем линии рамки в соответствии с требованиями ГОСТ. При рисовании линии (и не только линии, но и при работе с любыми объектами) нажатие клавиши **Tab** открывает панель редактирования свойств объекта. В данном случае нас интересует только толщина линии и ее тип. При проведении линий координаты можно задать явно, но быстрее воспользоваться заданием сетки и прорисовкой линий по ней. Сетка переключается циклически нажатием клавиши **G**. Установка, добавление, изменение сеток производится в настройках графического редактора **DXP/Preferences/Schematic/Grids**. Для выполнения надписей рекомендуется использовать шрифты, удовлетворяющие требованиям ГОСТ. Они находятся на сайте www.eltm.ru или www.platnaya.ru. Установите их на своем компьютере. В редакторе многие надписи используют только системный шрифт самого редактора. Для изменения системного шрифта используйте команду **Design/Document Option/Sheet Option/Change System Font**.

В соответствующих полях, как показано на рис. 2, введите текст согласно рекомендациям ГОСТ, причем для текста, который имеет переменное значение в зависимости от параметров схемы, введите знак «=» перед текстом. При заполнении полей можно использовать как стандартные поля, уже имеющиеся в редакторе по умолчанию, так и новые. Все новые поля вводятся в качестве параметров в свойствах документа (вкладка **Design/Document Option/Parameters**), где также можно определить их значение по умолчанию. Исключение составляют только поля, идентичные для первого и последующих листов многостраничных схем. Их после размещения в соответствующих графах рамки по требованиям ГОСТ следует удалить. Их затем проще ввести в параметрах проекта, и тогда они будут действовать на все листы его схем.

После завершения создания **Title** в настройках графического редактора схем восстановите галочку около **Convert Special String**. Рекомендуем выбрать на вкладке **Units** галочку около **Use Imperial Units System**, чтобы она была установлена по умолчанию. После этого надписи будут выглядеть так, как показано на рис. 2.

Аналогично создаются **Title** для других типов размеров листов.

При последнем сохранении **Title** желательно установить по умолчанию (вкладка **Design/Document Option/Units**) галочку около

Use Imperial Unit System. Для подключения **Title** схемы по умолчанию на вкладке **DXP/Preferences/Schematic/General** установите ссылку **Template** на одну из созданных **Title**, которая будет использоваться наиболее часто.

Создание библиотек

PCB Library

(TU0103 Creating Library Components)

Аналогично, как мы делали ранее, создадим новый **PCB Project** по команде **File/New/Project/Integrated Library** и сохраним его, например, в папку «..\Altium\Lib» с именем **Lib_GOST**. В данном проекте из контекстного меню по команде **Lib_GOST:Add New to Project/PCB Library** создадим новую библиотеку, которая будет содержать все посадочные места (**Footprint**) для наших первых проектов, и сохраним ее (**File/Save**) под именем **Example**. После чего данная библиотека появится в соответствующем проекте. Откроем ее и приступим к созданию **Footprint** нужных нам компонентов.

Для копирования элементов из другой библиотеки ее следует открыть как документ, выделить (через панель **PCB LIB**) нужные **Footprint**, скопировать в буфер (через контекстное меню), перейти в библиотеку, куда необходимо их скопировать и через контекстное меню панели **PCB LIB** вставить. Как правило, большинство **Footprint** не требует никакой корректировки, так как посадочное место физически отражает типоразмер. Процесс создания новых **Footprint** и редактирования старых понятен и не требует дополнительных комментариев. Для изучения нам вполне достаточно стандартных **Footprint**, поставляемых с приложением. Остановимся на нем ниже, после завершения создания первых проектов PCB только при создании нестандартных посадочных мест.

Schematic Library

Для создания схемы нам нужны компоненты схемных элементов, резисторы, конденсаторы и т. п. Как правило, основные элементы есть в библиотеках, поставляемых с приложением. Но, во-первых, они не всегда удовлетворяют требованиям ГОСТ, а во-вторых, всегда найдется такой компонент, которого нет в библиотеках, и его придется создавать. Рассмотрим процесс создания и редактирования новых компонентов.

Будем считать, что созданные нами компоненты пригодятся не только для схем, кото-

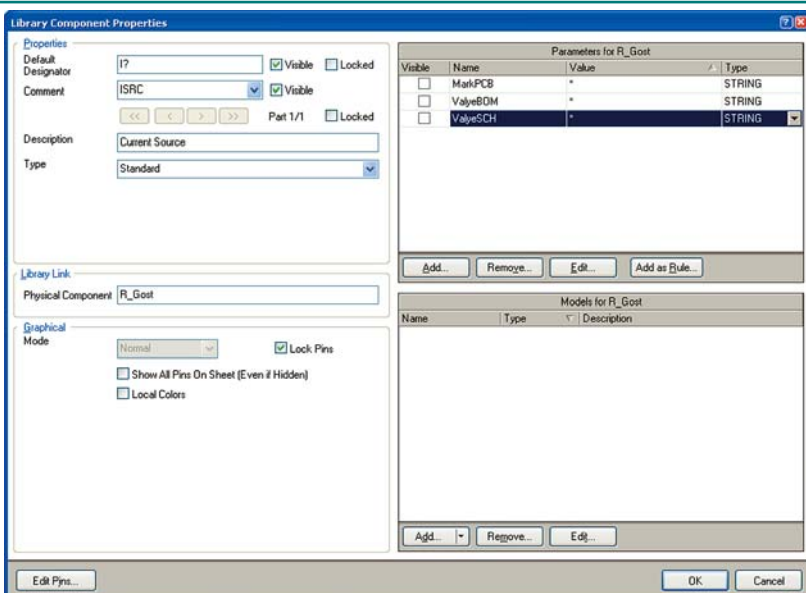


Рис. 3. Вид окна Library Component Property

рые будут рассмотрены ниже, но и для других проектов. Более того, со временем объем этих библиотек может стать достаточно большим, и их можно разделить, скажем, по функциональному назначению. Для простоты поиска компонентов эти библиотеки лучше сразу оформлять отдельным проектом.

Находясь в проекте **Lib_GOST** из контекстного меню по команде **Lib_GOST:\Add New to Project\Schematic Library** создадим новую библиотеку, которая, например, будет содержать все изображения резисторов, конденсаторов и индуктивностей, и сохраним ее (**File\Save**) под именем **RCL_GOST**.

При работе с библиотекой при редактировании ее компонентов активно используется панель **SCH Library**, и желательно, чтобы эта панель была всегда под рукой. Для этого лучше поместить панель **SCH Library** на выплывающую **Workspace Panels**. Открываем ее через **Access Panels\SCH\SCH Library**, перемещаем, к примеру, на левую сторону и делаем ее выплывающей (значок). После этого панель **SCH Library** легко вызвать нажатием соответствующей надписи на **Workspace Panels**.

1. Настройка параметров по умолчанию

Лучше сразу настроить эти установки, иначе затем вы создадите компоненты, цветовая гамма, линии и шрифты которых будут не единообразны. Затем придется тратить много времени на приведение их к идентичным формам. Сразу отметим, что все настройки вы не сможете установить, так как ваши вкусы могут поменяться со временем. Это тоже не страшно. Так как, владея уже более полно комплексом инструментов редактирования, вы позже сможете все исправить. Поэтому здесь мы укажем только тот минимум установок, которые нам надо изменить по требованиям ГОСТ.

Откроем окно **DXP\Preference:Schematic\Default Primitives**. Для **Primitives**, содержащих описание **Font**, а именно **Comment**, **Designator**, **Net Label**, **Note**, **Parameter**, **Sheet Symbol Designator**, **Sheet Symbol Filename**, **Text Frame**, **Text String**, установим **Font**= «**Gost Type A**, размер 12», что соответствует размеру

шрифта 2.5 мм, разрешенному ГОСТ. Рекомендуем для **Primitives=Part** ввести дополнительные параметры, а именно **ValueBOM** — значение, которое автоматически будет вписываться в графу «**Наименование**» при создании перечня элементов, **ValueSCH** — значение, которое будет отображаться на схеме, и **MarkPCB** — значение, соответствующее маркировке на корпусе компонента (для удобства формирования документов для проверки монтажа плат). Учтите, что все наименования параметров и их значения вы можете вводить и на русском языке. Но автор данной статьи — противник такого подхода, так как не теперь, так позже возникают проблемы с трансляцией проектов в другие пакеты, да и английские названия намного короче и точнее отображают саму суть.

2. Создание нового (удаление) компонента (TU0103 Creating Library Components) (рис. 3)

Откроем панель **SCH Library**. В списке компонентов находится один элемент — **Component_1**, создаваемый по умолчанию для новой библиотеки. Можно начать редактировать и его, но мы создадим новый (кнопка **ADD** в соответствующем окне). В появившемся окне **New Component Name** вместо **Component_2** вводим имя создаваемого нами компонента **R_Gost**, и приступим к созданию изображения резистора, в соответствии с требованиями ГОСТ. Но перед этим удалим не нужный нам в библиотеке **Component_1** (кнопка **Delete** в соответствующем окне).

3. Окно свойств компонента

Откроем компонент **R_Gost** (двойной щелчок мыши по его названию на панели **SCH Library**) и произведем следующие действия:

1. В поле **Default Designator** вводим значок **Visible** и надпись «**R?**», означающую, что в схеме данным компонентам будут присваиваться обозначения, начинающиеся с буквы **R**, а сами они будут отображаться на схеме. Знак «**?**» при операции **Annotate** уже в схемном редакторе будет автоматически заменяться индивидуальным позици-

онным номером в соответствии с заданными параметрами. Галочку около **Locked** следует снять, так как по умолчанию она означает запрет изменения обозначения и, как правило, если это требуется, устанавливается непосредственно в схеме. Это же касается и значка поля **Part 1/1 Locked**. Само поле нам недоступно и относится к компонентам, имеющим в отображении несколько частей. На нем мы остановимся при создании более сложных компонентов.

- В поле **Comment** выбираем из выпадающего списка надпись «**=ValueSCH**», которая означает, что на схеме будет отображаться значение приведенного параметра, и устанавливаем значок **Visible**. В данном поле можно оставить любой поясняющий текст. Но рекомендуем пользоваться специальными параметрами.
- Параметр **Description** используется только при поиске компонентов по библиотеке, он отображается в отдельном столбце (что позволяет облегчить поиск), а также при формировании **Report**. Так как одному компоненту, имеющему одно изображение, может принадлежать несколько физических **Part**, в данном поле введем просто «**Резистор двухвыводной**».
- В поле **Type** выбираем значение **Standard**, означающее, что данный компонент имеет соответствующий ему физический **Part** и будет внесен в перечень элементов.
- В поле **Physical Component** находится название компонента, под которым он хранится в библиотеке. Его изменять следует только в том случае, если вы хотите изменить название этого компонента в библиотеке.
- В поле **Graphical** устанавливаем значок **Lock Pins** и снимаем значки **Show All Pins On Sheet** и **Local Color**: их можно будет изменить, если понадобится, уже в схемном редакторе.
- В поле **Parameters for ...** мы видим три параметра, установленные по умолчанию. Все они нам не нужны для отображения на схеме, и галочку около **Visible** следует снять. Так как мы не знаем еще ни номинала компонента, ни его маркировки, значения всех параметров следует указать, например «*****». Дополнительно можете добавить параметры, описывающие, например, предельное напряжение, точность, температурный коэффициент, ссылку на описание, характерные для создаваемого компонента. Можно сделать их видимыми на схеме или скрытыми.
- В поле **Models for ...** можно подключить или удалить модели. Нас в данный момент интересует только **Footprint**. Одному компоненту можно назначить несколько **Footprint**. При этом в схемном редакторе появится возможность их выбора.
- Нажав кнопку **Edit Pins** можно перейти в окно редактирования **Pins** компонента. Однако этим лучше пользоваться уже при сформированном изображении компонента. Рассмотрение этого окна будет приведено ниже.
- Редактор компонента**
Перейдем на поле открытого компонента. Оно пока пустое.

С помощью команд **Place Pin**, **Place Line** и т. п. панели **Utilities** создадим изображение резистора в соответствии с требованиями ГОСТ. Единственная рекомендация — сделать привязку **Pin** в сетке 100 mil, что не противоречит требованиям ГОСТ, но облегчит применение других элементов из стандартных библиотек, а также передачу данных в другие приложения, которые поддерживают только дюймовую метрику. Выделив **Pin** и нажав клавишу **Tab** на клавиатуре, открываем окно **Pin Property**. Заполнение свойств **Pin** обычно не вызывает затруднений (рис. 4).

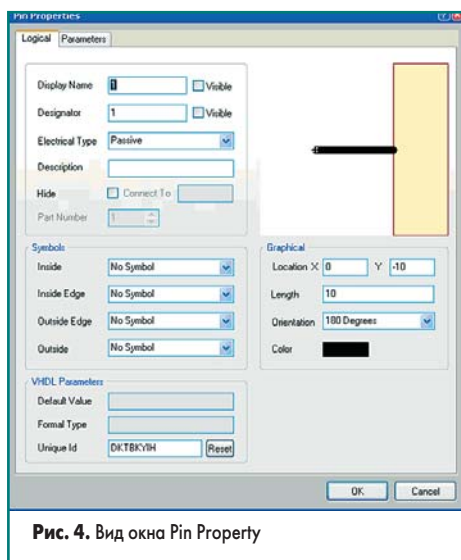


Рис. 4. Вид окна Pin Property

Для ввода компонентов, содержащих два и более независимых частей (это, как правило, логические элементы или компоненты с большим числом пинов, разделенные на функциональные модули), используется команда **Add component part** на панели **Utilities**.

Для одного компонента можно сделать несколько альтернативных изображений. Сделаем для нашего компонента второе изображение, с обозначением, например, мощности. Выделим изображение нашего компонента и скопируем его в буфер, при помощи команды **Add a display** на панели **Utilities** откроем окно редактирования альтернативного изображения, вставим изображение из буфера и отредактируем его, внося графическое обозначение мощности.

С одним компонентом может быть соотносено несколько наименований изделий, например, для нашего резистора **R_Gost** сопоставим ряд резисторов как R1206, R0805 и т. п. Для этого используется команда **Add** окна **Aliases** на панели **SCH Library**.

Аналогичным образом создадим и остальные компоненты, которые понадобятся для схемы. Отредактировав элемент в библиотеке, вам не нужно будет делать это для следующего проекта.

При создании изображений компонентов следует соблюдать ряд полезных советов, чтобы избежать некоторых недоразумений, которые могут возникнуть впоследствии.

1. При вводе новых параметров лучше копировать их из уже существующих компонентов, это снижает ввод различных по напи-

санию параметров, несущих одну функциональную направленность.

2. Любая надпись (например, функциональное назначение микросхемы), предназначенная для редактирования в схеме, должна быть введена как параметр.

3. При применении закрашенных фигур (прямоугольников, овалов и т. п.) последним действием должен быть их перевод на нижний слой командой **Edit\Move\Send to Back**. Это позволит тождественно передать изображение схемы в PDF-формат.

5. Изменение параметров, не отображаемых в редакторе

Если вы редактировали компонент, взятый из другой библиотеки, возникает необходимость изменения таких его параметров, как **Designator** и **Comment**, для того чтобы не править их потом в схеме. В редакторе компонентов эти два параметра не отображаются, поэтому их редактирование иногда вызывает сложность. Решить эту проблему можно следующим образом.

1. Вызываем панель **SCHLib List**.

2. Устанавливаем режимы: редактирования **Edit**; выделения всех элементов **all object** для текущего компонента или всех компонентов открытых библиотек; включаем **all types of object**.

3. Выбираем в столбце нужные нам параметры, например **Designator**, после чего в соответствии с предыдущим пунктом выбираем режим выделения **selected object**.

4. После этого в соответствующих столбцах изменяем параметры **Designator**.

Аналогично можно править не только эти, но и другие параметры любых объектов компонента, всех компонентов проекта или открытых документов.

Int Library

Для того чтобы создать интегральную библиотеку, содержащую несколько библиотек схемных изображений, корпусов и их моделей, объединенных в одном проекте, используется команда **Project\Compile...** Как правило, объединению подвергаются проверенные библиотеки. Интегральную библиотеку вы можете проинсталлировать и использовать так же, как и индивидуальные библиотеки. Запомните, перенос изменений отредактированных элементов производится только после операции **Project\Recompile...**

Схемный редактор (AP0109 Schematic Editing Essentials)

Создание листа новой схемы в проекте

Приступим непосредственно к созданию схемы.

Аналогично предыдущему разделу создадим:

- новый проект **PCB Project** под именем **Example** и сохраним его в директории **..\Altium\Project\Example**;
- новый документ в проекте **Example_1** типа **Advanced Schematic ascii** и сохраним его в директории **..\Altium\Project\Example**.

Хотя эту операцию можно произвести и по-другому: нажав правую кнопку мыши, находясь в соответствующем проекте. При этом откроется заготовка схемы со встроенной **Title**. Вы всегда на любой стадии редактирования документа можете изменить или удалить **Title**, чтобы схема входила на нужный типоразмер, командой **Design\Template\Set Template File Name**, а также заполнить все поля **Title** в соответствии с создаваемой схемой на вкладке **Design/Document Option**, аналогично как вводили при создании **Title**. Нужно ввести общие поля для всех листов и задать значение **Project/Project Option/Parameters**.

Итак, практически все готово к созданию самой схемы.

Внесение в проект элементов схемы

В качестве примера создадим схему DC/DC преобразователя, взятую из DataSheet LM5008 (www.national.com) (рис. 5).

Все объекты на схеме делятся на две группы: графические и электрические.

Для размещения графических объектов (линий, дуг, текста и т. п.) используют панель **Utilities**. Для размещения электрических объектов (электрических связей, наименований цепей или групп цепей, компонентов и т. п.) используют панель **Wiring**.

Основным объектом являются библиотечные компоненты. Вначале попытаемся найти их в библиотеках, поставленных с приложением **Altium Designer**. Так как работа с библиотеками является постоянной, желательно, чтобы они были всегда под рукой. Для этого лучше поместить панель **Library** на выплывающую **Workspace Panels**. Открываем командой **Access Panels\System\Library** панель **Library** и перемещаем ее, скажем, на правую

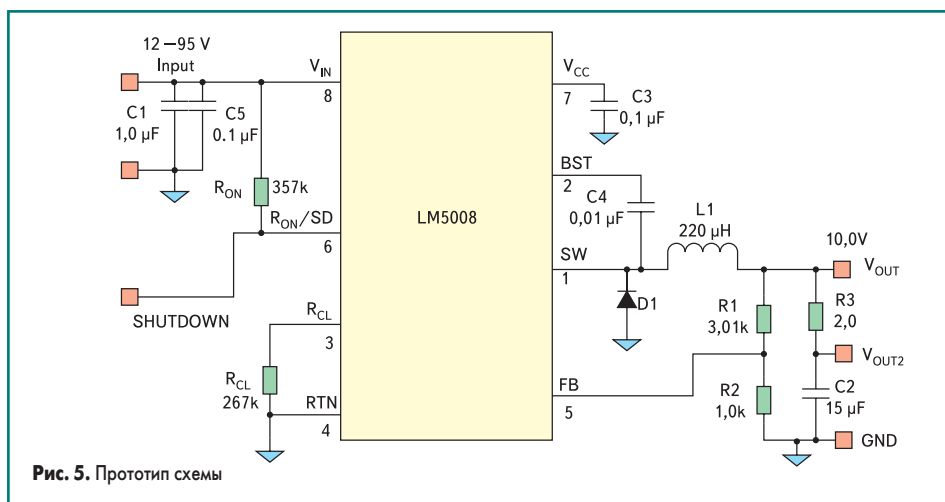



Рис. 5. Прототип схемы

сторону и делаем выплывающей (значок ). После этого панель **Library** легко вызвать нажатием соответствующей надписи на **Workspace Panels**.

Теперь попробуем найти элемент LM5008 в существующей библиотеке. Для этого на панели **Library** нажимаем кнопку **Search**. В появившемся окне **Library Search** пишем название нашего компонента, и так как мы знаем, какая компания его выпускает, ставим галочку около **Library on patch** и выбираем директорию производителя. Это желательно сделать для сокращения времени поиска, так как библиотек очень много и поиск по всем библиотекам занимает большой период времени. Зная описание библиотек, можно еще сократить время, задав маску поиска по файлам **File Mask**, но это уже лишнее. Поиск пройдет и так быстро. После чего нажимаем кнопку **Search**.

Наш поиск оказался безуспешным. В этом случае есть несколько продолжений:

- произвести поиск в Интернете на сайте производителя микросхемы или разработчика приложения www.altium.com;
- произвести поиск по аналогам других производителей;
- выбрать и вставить простые элементы из панели **Utilities**;
- создать компонент самому.

Первый способ понятен. И если найдете там нужную библиотеку, скопируйте ее в соответствующую директорию или создайте для нового производителя свою директорию. Во втором случае задайте поиск, например, по части текста **5008** и по всей корневой директории **Library**. Поиск занял существенно больше времени и нужных нам компонентов не дал. Не стоит отчаиваться, у нас есть третий способ. И он во многих случаях и будет основным. Но на нем остановимся чуть ниже. А пока аналогичным способом поищем резисторы и конденсаторы по названию 1206. таких элементов будет найдено несколько, но, как правило, они не соответствуют требованиям нашего ГОСТ, и его проще сделать самим. Но если вы все же захотите внести его в схему, нажмите кнопку **Place...** на панели **Library**. Так как данная библиотека не установлена, вам предложат ее проинсталлировать. Это вам, конечно, решать, но лучше отказаться. Иначе в больших проектах вы потеряетесь во всех многочисленных проинсталлированных библиотеках. Лучший способ, по мнению автора, создавать библиотеку для каждого проекта. Или собственные библиотеки — из созданных вами хоть один раз или скопированных из других библиотек компонентов.

Выбрать и вставить наиболее часто используемые элементы можно и из панели **Utilities** — это один из простых методов (рис. 6). И действия там очевидны из рисунка. Поставим с этой панели заготовки резистора, конденсатора и на лист нашей схемы. Однако по умолчанию они подключены к стандартным библиотекам. Изменить наиболее часто используемые элементы в этой панели или добавить новые можно, открыв окно **DXP\Customs\Utilities** и отредактировав строку **Parameters** соответствующего значка. Однако делайте это, только чувствуя себя опытным пользователем.



Рис. 6. Вид элементов на схеме и вставка их через панель Utilities

Составим схему на основе созданных нами элементов. Проинсталлируем библиотеку для всех проектов, которые мы разработаем позже (панель **Library**, кнопка **Library** и поставим ее первой в списке (рис. 7). Можете сразу установить (или удалить) и другие библиотеки, впоследствии созданные вами или поставляемые с приложением.

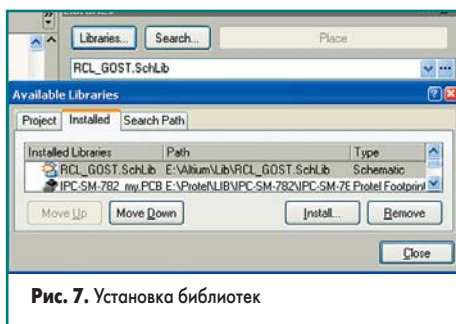


Рис. 7. Установка библиотек

Добавить компоненты на схему можно следующими способами:

- из панели **SCH Lib**, выделив компонент и нажав кнопку **Place**. При этом оба документа, библиотека и схема должны быть открыты;
- из панели **Library** выделением и «перетаскиванием» на схему нужных компонентов из установленных библиотек;
- непосредственно из схемы **Place\Part** или панели **Utilities** кнопка **Place Part** дает доступ к недавно использованным компонентам, ко всем компонентам установленных библиотек, к поиску компонентов в любых библиотеках;
- копированием компонента, уже имеющегося на схеме (достаточно при выделении элемента мышкой нажать кнопку Shift и «перетащить» копию в нужное место).

Размещение на схеме других графических и электрических объектов для нашей простой схемы не должно вызывать затруднений.

Добавление или изменение атрибутов компонентов на схеме

Ввести или изменить в схеме надписи, относящиеся к конкретному элементу, можно вы-

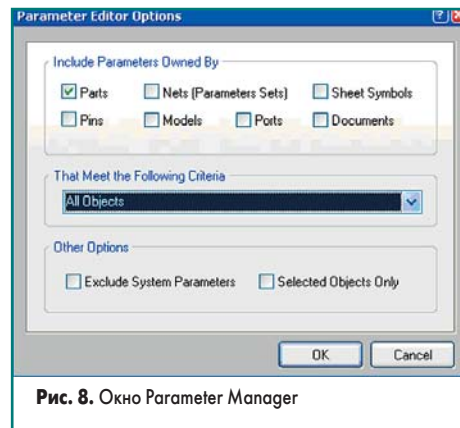


Рис. 8. Окно Parameter Manager

делив его и не отпуская кнопку мышки нажать Tab. В появившемся окне вводим или правим необходимые атрибуты (например, ValueSCH — для отображения на схеме номинала компонента). Однако такой метод хорош для уже созданной схемы, в которую нужно ввести только некоторые коррективы, или для очень простых схем. Для вновь созданной схемы проще использовать команду **Tools/Parameter Manager**.

В появившемся окне установите галочки, как показано на рис. 9, так как мы будем править только те параметры, что принадлежат компоненту (Part), введенные нами, и нажимаем OK.

В данной таблице находится три области:

- информационная — указывается общее число компонентов и приводится список типа объекта, его нахождение в проекте и позиционный номер. Эта область недоступна для редактирования;
- System parameters — системные параметры (зарезервированные наименования для ввода моделей, описаний, ссылок и т. п.). Системные параметры можно изменить, но не удалить;
- User parameters — пользовательские параметры, которые могут быть добавлены, удалены, изменены.

Как правило, начинающие пользователи уже при создании библиотечных элементов создают множество пользовательских параметров, дублирующих друг друга или не несущих информации. Для того чтобы удалить столбцы с параметрами, которые мы не будем использовать, нужно столбец выделить и воспользоваться контекстным меню для удаления или редактирования. Так же можно и добавить новые, или переименовать старые. Для изменения параметра нужно выделить одну ячейку или диапазон ячеек и ввести или вставить

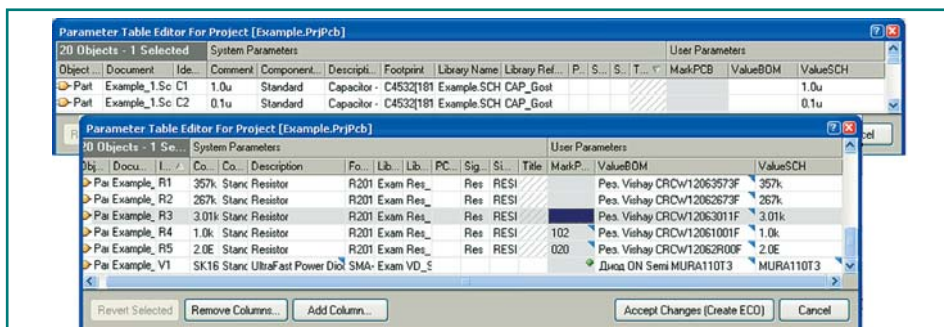


Рис. 9. Окно Parameter Table Editor For Project. До и после редактирования параметров

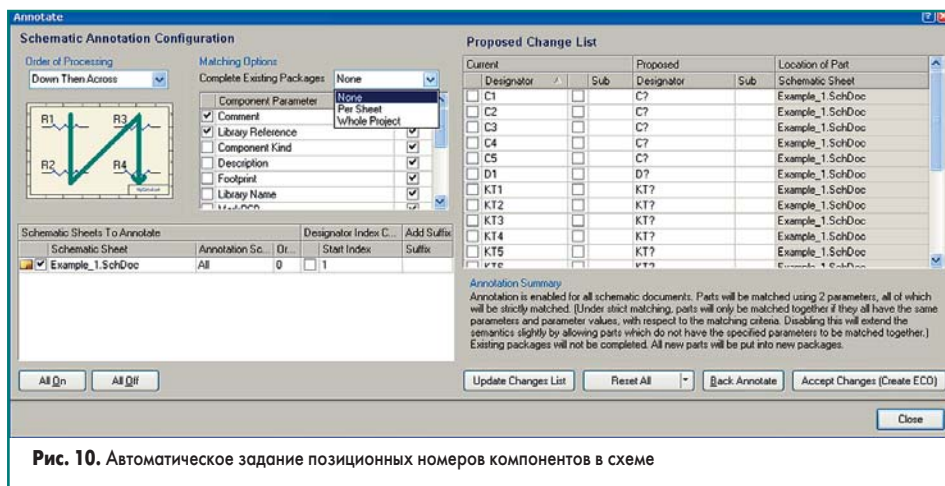


Рис. 10. Автоматическое задание позиционных номеров компонентов в схеме

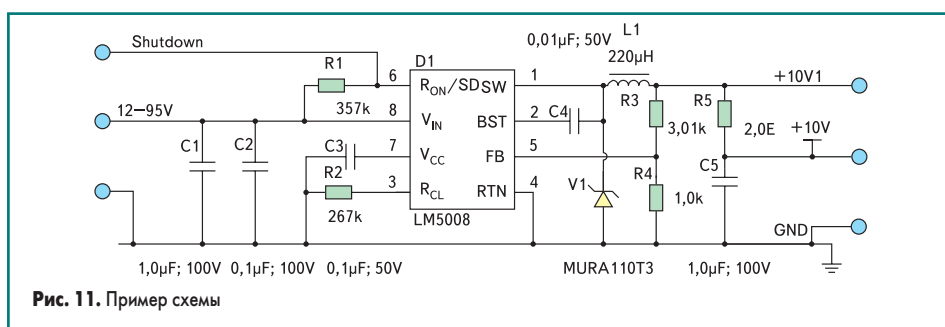


Рис. 11. Пример схемы

из буфера обмена требуемое значение. После всех изменений синим треугольником будут обозначены измененные параметры, знаком «+» — введенные впервые для компонента. Для того чтобы принять изменения нажмите кнопку **Accept Changes (Create ECO)** (формирует файл внесения изменений), затем **Validate Changes** (для проверки возможности выполнения действий) и кнопку **Execute Changes** (непосредственный ввод изменений в схему).

Присвоение позиционных номеров компонентам схемы

После размещения всех объектов схемы сначала пронумеруем элементы. Для этого используем команду **Tools/Annotate** (AP0109 Schematic Editing Essentials.pdf).

В появившемся окне задается:

- **Order of processing** — выбор направления автоматического присвоения позиционных номеров (рис. 10);
- **Matching Option** — порядок группирования в один корпус гетерогенных элементов. При этом можно задавать дополнительные условия для группирования: по заданным листам и по заданным параметрам компонентов;
- **Schematic Sheet To Annotate** — выбор компонентов листов схем для аннотации, в том числе:

- **Schematic Sheet** — выбор листов схем для аннотации;
- **Annotation Scope** — выбор компонентов в листе для аннотации;
- **Order** — задание номера листа для многолистовых схем;
- **Start Index** — задание начального номера для нумерации;
- **Suffix** — добавление суффикса при нумерации;
- **Reset ...** — сброс всех позиционных номеров или только дубликатов.

Выбираем необходимые условия и нажимаем кнопку **Update Change List** (формирует подготовку к изменению).

На схеме редактируем положения всех надписей так, что бы они не мешали друг другу.

В результате получим схему, как показано на рис. 11, которая соответствует исходной. Мы не старались полностью удовлетворить всем требованиям ГОСТ, так как стремились минимизировать размеры схемы — для журнального варианта.

Выбор и изменение модели Footprint-компонента

Когда создавали библиотеку, мы еще не знали, какие корпуса будут иметь наши ком-

поненты (особенно это касается резисторов, конденсаторов) и подключили только самые распространенные корпуса. Настало время определиться с этим для созданной схемы. Войдем в свойства компонента. Выбор **Footprint** производится в выпадающем меню **Name** окна **Model for...** В списке присутствуют только те **Footprint**, которые мы определили в библиотечном элементе. Если нас он не устраивает, его можно удалить или назначить новый (кнопка **Edit**). Это весьма примечательно, так как не нужно привязываться к моделям, создавая библиотеку, а только задавать их в схеме.

Однако если нужно править **Footprint** для большого числа элементов схемы, лучше воспользоваться командой **Tools/Footprint Manager** (рис. 12). Однако перед этим следует установить все библиотеки **Footprint**, которые мы будем использовать.

Итак, для конденсаторов C1 и C2 это типоразмер 4532. Выделяем две строчки с этими компонентами, нажимаем кнопку **ADD** и производим поиск **Footprint** с именем «4532». Поиск обнаруживает несколько **Footprint** содержащих в названии «4532». Выбираем тот, который находится в стандартной библиотеке. Если **Footprint** найден, его изображение сразу появляется в соответствующем окне. Аналогично поступаем с остальными компонентами. После подключения **Footprint** все готово к созданию PCB.

Завершение формирования схемы

Считаем, что наша схема завершена. Теперь необходимо ее скомпилировать. Для этого используется команда **Project/Compile...** Можно скомпилировать только активный документ или для многолистовых схем — весь проект. После компиляции проекта в панели **System:Messages** при наличии ошибок появится сообщение о них. В нашем случае схема не содержит ошибок, и панель **Messages** пуста. Для того что бы получить ошибку, поставьте на одну цепь, например, **Shutdown**, еще один «Net name» = «1». И в данной панели возникнет сообщение (рис. 13).

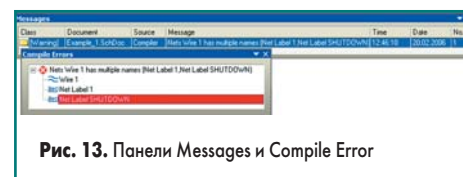


Рис. 13. Панели Messages и Compile Error

Внимательно отнеситесь ко всем сообщениям панели **Messages** и устраняйте принципиальные ошибки. При двойном щелчке по сообщению появляется окно **Compile Error** с детализацией сообщения. При двойном щелчке по сообщению **Compile Error** в основном поле открывается документ, содержащий эту ссылку, и выделяются соответственный элементы схемы, а остальные маскируются. Это все позволяет достаточно оперативно проанализировать все сообщения.

Для ряда ошибок достаточно сложно найти причину. Но на характерных ошибках мы остановимся в отдельном разделе. Теперь же приступим к передаче данных в PCB-проект.

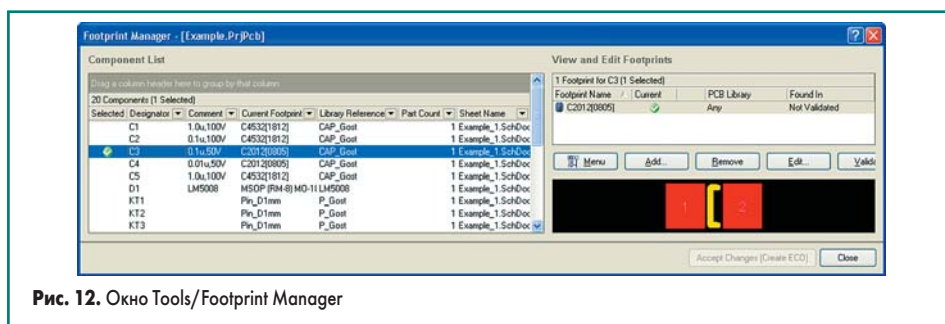


Рис. 12. Окно Tools/Footprint Manager

Продолжение следует

Продолжение. Начало в № 5 '2006

Система проектирования Altium Designer 6

Владимир Пранович

Pranovich@bsu.by

Параметры проекта

Перед передачей данных в PCB-проект желательно задать параметры проекта в целом, так как от этого зависят условия передачи данных. Командой **Project\Project Option** открываем окно параметров проекта. Мы оставим все установки заданными по умолчанию. Однако, чувствуя себя опытным пользователем, вы можете их установить по вашим требованиям. Те изменения, которые нужно будет делать при создании сложных проектов, будут указывать в соответствующих разделах.

На вкладке **Error Reporting** можно задать проведение контроля на соответствующие поля и степень ошибки. Число параметров, по которым проводится проверка, очень велико, и не имеет смысла их здесь приводить. В любом случае помните, что лучший контролер — это вы сами.

На вкладке **Connection Matrix** аналогично задается проверка схем на целостность сигналов.

На вкладках **Class Generation** задаются методы образования классов **NET** и **ROOM** для компонентов на PCB.

На вкладке **Comparator** задаются правила нахождения различий между схемой и PCB.

На вкладке **ECO Generator** можно включить или отключить формирование **ECO** для отдельных классов отличий.

На вкладке **Option** задаются порядок хранения документов проекта, правила связи иерархических схем, а также функции **SWAP**.

На вкладке **Multi-Channel** задается порядок имен **NET** и позиционных номеров для иерархических и сложных схем.

Вкладка **Default Prints** — вид документов и их наполнение при формировании отчетов и выходных файлов. Удобная вещь: когда вы подготовите уже не один документ, то сможете быстро формировать выходные файлы. Это же касается и вкладки **Search Patch**.

На вкладке **Parameters** зададим параметры, общие для всех листов проекта. Например, таким параметром является номер разработки **Dec_N = EXAP.EXAMPLE.000.01 Э3**. Для PCB нужна только общая часть **Dec_N**, и для нее вводим параметр **Dec_N = EXAP.EXAMPLE.001.01**. Для того чтобы общая часть передавалась в схему, проверьте, чтобы у конкретного листа схемы не был прописан собственный параметр **Dec_N**. Как правило, его забывают удалить при создании **TITLE**, и он кочует из схемы в схему. Если он есть, его следует удалить.

PCB редактор (TU0117 Getting Started with PCB Design)

Добавим к проекту новый PCB-документ и сразу сохраним его под именем **Example**.

В начале создадим заготовку проекта, которая в дальнейшем может служить **Template** при создании новых PCB-проектов.

Создание рамки

Создадим в слое, например, **Mechanical16** рамку, идентичными методами, как и для схемы. Учтите, общие названия с рамкой из схемы могут быть автоматически переданы из проекта через его параметры. Для этого при вставке текста перед параметром ставится точка (рис. 14). Для задания рабочего поля (на весь размер выбранного нами листа) используйте команду **Design\Board Shape\Redefine Board Shape**.

После этого его можно сохранить в качестве **Template** в соответствующую папку.

Передача данных между схемой и PCB и создание простых проектов

Передача данных из схемного редактора в PCB производится командой **Design\Update PCB....** Эта команда идентична импорту данных в редакторе PCB из **Design\Import Change ...** из исходного проекта

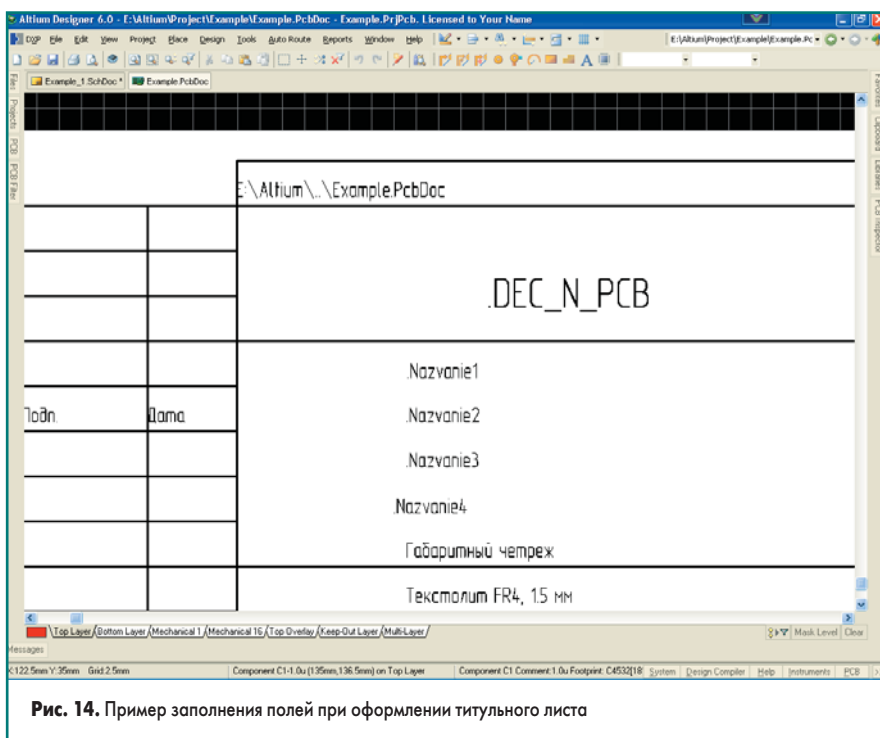


Рис. 14. Пример заполнения полей при оформлении титульного листа

(рис. 15). Для передачи изменений, сделанных в PCB-редакторе в схемный редактор, используется команда **Design\ Update Schematic ...**. Пока нам понадобятся только первые две. Связь между компонентами схем проекта и PCB производится по уникальным номерам. В результате многократных изменений в схеме или PCB, связь между проектами может быть нарушена. Для изменения задания связи между компонентами схемы и PCB при активном PCB-документе используется команда **Project\Component Link ...**. В первых двух колонках указаны компоненты схемы и PCB, связь между которыми не установлена, справа — связанные компоненты. Вы легко можете устранить связь, ввести ее вручную или автоматически, указав критерии связи по параметрам (**Designator**, **Comment**, **Footprint**). Следует всегда следить за однозначностью связи, но при любой передаче данных, если отсутствует такая связь, всегда будет выдано предупреждающее сообщение.

Из схемы в PCB-проект всегда передаются параметры **Designator**, **Comment**. Установив для параметра **Comment** значение, например, **=ValueSCH**, можно передать соответствующий параметр. Для желающих иметь одновременно не один, а несколько параметров компонента в PCB-проекте, при создании Footprint следовало добавить текст, например, **.Designator** или **.Comment** указав слой и параметры шрифта. Однако рекомендуем этот прием использовать только чувствуя себя уверенным пользователем. Тем более, что большой необходимости в этом нет.

Контур будущей платы задается линиями и окружностями в одном из слоев **Mechanical**, область, доступная для разводки в слое **Keep Out**. Если они совпадают, можно ограничиться последним. Начнем с простого прямоугольного контура. Переключение между слоями производится нажатием «*» на дополнительной клавиатуре или нажатием на закладку соответствующего слоя. Если вы этого слоя не видите, включите его из окна «Контекстное меню \Option\Board layer & colors...» (рис. 16). Здесь же вы можете настроить цвета, добавить слои и т. п. После создания контура платы следует всем элементам контура установить свойство **Locked**, чтобы избежать случайных перемещений элементов контура.

Перейдем теперь к компонентам. У нас плата маленькая. В ней образована одна **Room** под именем **Example_1**, совпадающим с именем листа схемы, так как мы использовали установки по умолчанию. Чтобы поместить/удалить компоненты внутри **Room**, используйте команды **Tools\Component Placement\Arrange Components Within Room, ...\Arrange Components Outside Board**. Перемещая **Room**, мы перемещаем и все компоненты. Нам проще переместить и изменить формы **Room** по размерам будущей платы, сделать для **Room** опцию **Room Locked**, а затем переместить все компоненты внутри **Room**. После этого у нас все элементы стоят, как на рис. 17. Для пользователей с любой системой проектирования плат, не составит труда получить и конечный продукт разводки.

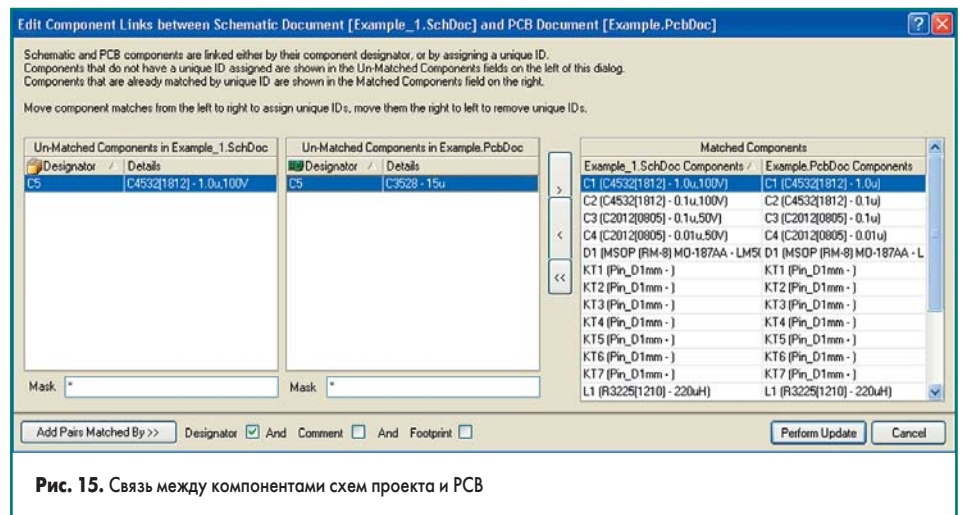


Рис. 15. Связь между компонентами схем проекта и PCB

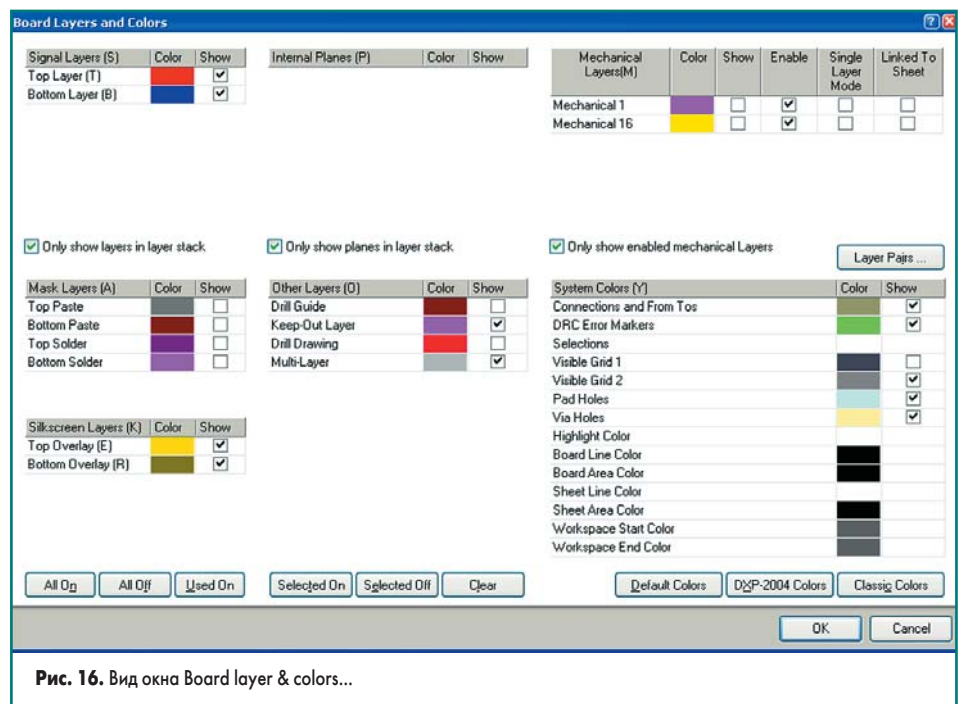


Рис. 16. Вид окна Board layer & colors...

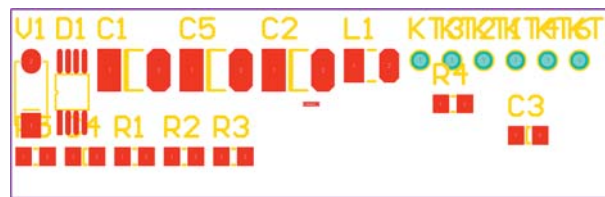


Рис. 17. Начальное расположение элементов на Board

Отметим только ряд действий, которые придется совершить:

1. Установленный по умолчанию параметр **Clearance** имеет значение 0,254 мм. Это не позволит провести дорожки, например, к корпусу MSOP (RM-8) MO-187AA. Для изменений этого параметра используется команда **Design\Rules\Electrical+Clearance**, где наглядно устанавливается данный параметр. Делаем его значение равным 0,2 мм. Здесь сразу же можете ознакомиться с большим числом правил при проектировании плат. Определить их значения и есть одно из главных действий для опытных пользователей.
2. Аналогично устанавливаем параметр **Design\Rules\Routing+Width** (минимальное значение

— 0,15 мм, по умолчанию — 0,25 мм, максимальное — 0,5 мм).

Будем считать процесс создания трассировки завершенным (рис. 18).

Наиболее часто используемый способ при проектировании — это возможность установки на плату одного из нескольких вариантов корпуса компонента. При этом речь не идет о выборе одного из вариантов корпуса (этот вопрос легко решаем и не вызывает затруднений), а о соответствии компонента сложному посадочному месту, включающему одновременно несколько вариантов исполнения корпуса, например для SMD элементов корпуса типоразмеров 0805 и 1206, для транзисторов — совмещение корпусов SOT23, SOT223, DPACK,

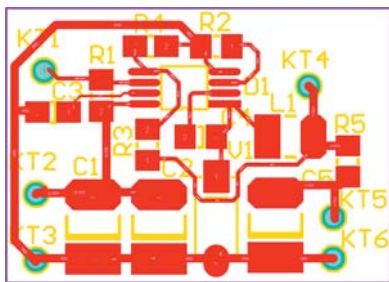


Рис. 18. Пример разводки

для микросхем — SOIC и DIP. Остановимся на создании составного корпуса для микросхемы LM5008, включающего две разновидности MSOP8 и LLP8.

Вернемся в редактор **PCB Library (TU0103 Creating Library Components)** и дополнительно к существующему посадочному месту MSOP8, которое мы уже использовали, создадим LLP8. Затем создадим совмещенный корпус MSOP8_LLП8 копированием в него изображений его составляющих. Заметьте, что новый корпус будет содержать по два PAD с номером 1, по два PAD с номером 2 и так далее. В слое **Top Layer** командой **PlaceLine** соединяем PAD с одинаковыми номерами. Новый компонент готов. Таким же образом можно задать сложную форму для контактной площадки, положив сверху полигон. Но при использовании полигона, если требуется освобождение его от маски, соответствующая графика должна быть прорисована отдельно, так как для полигонов, в отличие от PAD, автоматической генерации ее нет.

Теперь подключаем новый компонент в качестве текущего в схему (напрямую, или введя соответствующие изменения в **Schematic Library**) и обновляем PCB-проект.

К сожалению, изображение данного составного корпуса не совсем информативно, так как корпус LLP8 можно смонтировать на месте посадочного места MSOP8. Поэтому на рис. 19 показано также и составное посадочное место SOT23_SOT223_DPACK. В последнем примере PAD корпуса SOT23 выполнены полигонами и соединены линиями с PAD корпуса DPACK; PAD корпуса SOT223 совмещены.

Подготовка документации и файлов для производства

Для подготовки документации в пакете имеется развитая система их автоматического формирования. Рассмотрим только основные из них. Остальные легко осваиваются по мере надобности.

Формирование файлов для производства печатных плат

Команда **File\Fabrication Output\Gerber Files** используется для создания Gerber файлов.

- В окне **General** задаем размерность и формат формируемых файлов.
- В окне **Layers** задаем слои, для которых формируются выходные данные. В нашем случае (считаем, что плата однослойная) это:
 - **Top Layer** — верхний слой;

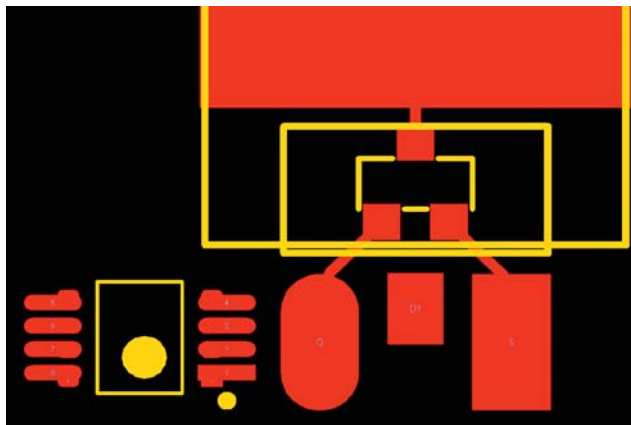


Рис. 19. Примеры составных посадочных мест

- **Top Overlay** — верхний слой шелкографии (надписей);
- **Keep Out Layer** — контур обрезки.
- В окне **Drill Drawing** происходит задание отображения символов сверловки. Мы будем формировать файл сверловки отдельно. Поэтому можно их не задавать.
- В окне **Apertures** — задание апертур. Ставим галочку **RS274X**.
- В окне **Advance** — размеры фотошаблонов. Оставляем по умолчанию.

При оформлении документации можно прикрепить к формируемым **Gerber Files** дополнительный слой, скажем, с нашей рамкой, выполненной по ГОСТу

Команда **File\Fabrication Output\NC Drill Files** используется для формирования файла сверловки.

Формирование сборочного и монтажного чертежа

Команда **File\Assembly** используется при формировании вспомогательных файлов для удобства монтажа печатных плат. Предварительно все слои и тип выводимой информации задаются в окне **Project\ProjectOption\Default Prints:Assembly Output: Assembly Drawing**. Здесь же можно задать и вывести на печать большое количество созданных вами заготовок для документов.

Рассмотрим подробнее дополнительные изменения в РСВ-проекте для удобства дальнейшей работы и соответствия требованиям технологических чертежей.

Для подготовки сборочного чертежа для монтажного участка расположение и распечатка позиционных обозначений элементов, используемых в слое **Top Layer**, не совсем удобна. Во-первых, надписи в этом слое могут иметь малый размер и при выводе на печать во-вторых, при плотном монтаже они вообще могут не насены на чертеж или плату, и в-третьих, что самое принципиальное, они могут иметь не адекватное прочтение, что вызывает некорректную сборку печатной платы. Эту проблему можно решить двумя способами:

- Первый. В редакторе PCB выделим все компоненты и установим свойства для параметра **Designator** — размещение (**autoposition**) в центре и нужный слой — и дальше готовим вариант для печати. Это самый простой и очевидный способ, но у него есть несколько

ко недостатков. Во-первых, необходимо делать «откат» от произведенных действий, чтобы не изменить надписи в слое **Top Layer** (изображение надписей на плате не должно меняться), а про это часто забывают. Во-вторых, надписи для удобства чтения лучше сделать максимально большими — по размеру корпуса компонента.

- Второй способ может показаться сложнее. Зато он в принципе позволяет получать независимые изображения **Designator** для **Top Layer** и другого технологического слоя, который и может использоваться при создании чертежей для монтажного участка. Более того, таким же способом можно получить и другие чертежи, например, с маркировкой элементов или их номиналами.

Для этого вернемся в редактор *PCB Library* (*TU0103 Creating Library Components*).

Здесь для всех компонентов командой **Place\String** ввести на слой, который будет использоваться при изготовлении чертежа для монтажного участка, например, **Mechanical 1**, надпись **.Designator** и выбрать положение шрифта и его размер в соответствии с конкретным корпусом. Будьте внимательны! Именно точка в начале надписи означает, что данные будут передаваться из схемы в PCB-проект. Затем все компоненты в PCB-проекте следует обновить **Tool\Update...** в соответствии с измененной библиотекой и установить галочку **Convert Special String** в установках PCB-редактора (**DXP\Preferences\PCB Editor\Display**). Далее останется только провести редактирование тех надписей, которым присвоены значения **Designator**, состоящие из длинных записей и не вписывающиеся из-за этого в размер корпуса. При грамотном подходе эти надписи править не нужно, и для того, чтобы они были закрыты по умолчанию, для редактирования в свойстве надписи (в **PCB Library**) следует установить свойство **Locked**. На рис. 20 приведены рисунки исходного вида чертежа части платы до и после проведения указанных действий.

Таким образом, можно из схемы в РСВ-проект передавать и значения любых других параметров, например, ранее введенный нами параметр **MarkPCB** (маркировку, нанесенную на корпус элемента). Однако, здесь есть ограничение. Для этого при создании по-



Рис. 20. Вывод Designator в дополнительном слое

сачдного места используется параметр **Comment**, а для передачи значения другого параметра из схемы на PCB используется его переопределение **Comment=MarkPCB**. Это удобно для поиска и определения типа смонтированных компонентов, так как на малых по размеру корпусах наносят несколько символов, по которым можно найти полное название элемента.

Формирование заготовки перечня элементов схемы и спецификации сборочного чертежа

Для этого используется команда **Reports\Bill of Materials**. Она доступна и из схемного редактора, и из редактора PCB. Типовые настройки показаны на рис. 21. В колонку **Grouped Columns** из колонки **All Columns** переместите параметры, по которым должно производиться группирование, при создании спецификации. В обеих этих колонках отметьте галочками те параметры, которые будут присутствовать в спецификации. В правом окошке перетаскиванием колонок создайте их порядок следования в спецификации. Внизу укажите формат сохраняемого файла. Удобно сразу подключить файл, содержащий заготовку спецификации. В этом случае вы получите практически готовый перечень элементов или спецификацию. При использовании EXCEL есть одно ограничение. Для больших проектов надписи, например в колонке **Designator**, могут не поместиться в ячейку EXCEL, и тогда они игнорируются. На рис. 21 выведены установки для стандартных колонок в перечне элементов.

Формирование полного пакета выходных данных

Так как описано выше, вы можете создать указанные документы для вашего проекта. Однако каждый раз формировать и прописывать параметры формируемых документов нет необходимости. Достаточно один раз настроить их для типовых проектов и затем использовать как стандартные. Для этого введем в проект новый документ **Project\Add New to Project\Output Job File**. В открывшемся документе вы можете подготовить любые формируемые выходные файлы и их параметры, включая различные модификации. В том числе для схем, поддерживающих наличие раз-

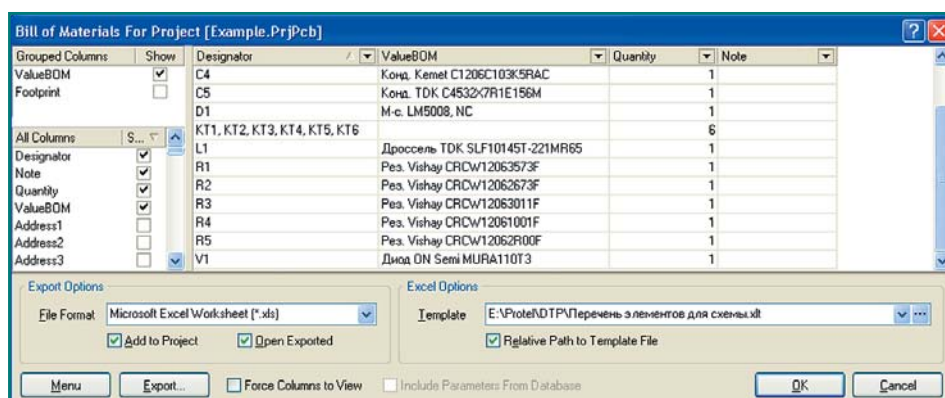


Рис. 21. Создание перечня элементов

личных элементов и номиналов, в зависимости от исполнения изделия. Один раз прописанный файл типа **Output Job File** может быть легко скопирован или добавлен к новому проекту, это дает существенное сокращение времени на подготовку выходных файлов для каждого нового проекта.

Иерархические схемы

Перейдем к одному из принципиальных отличий **Altium Designer 6** от **PCAD200x** — это возможность построения иерархических схем и автоматический перенос трассировки идентичных блоков в PCB-проект.

Требования к иерархическому проекту и подготовка к его созданию

Итак поставим себе задачу создать проект, включающий в себя 6 DC/DC преобразователей, схема которых нами использовалась в предыдущем проекте, на формирующие

уровни питания: один на 3,3 В, четыре на 5,0 В и один на 10 В.

Как уже делали ранее, сформируем новый проект **Example2**. Хотя проект можно перенести и обычным копированием. Единственное необязательное требование: дайте всем файлам проекта и самому проекту другие имена, откройте его и добавьте к новому проекту переименованные документы. Откроем пока единственный лист нашей схемы и сохраним (перенесем) в новом проекте под именем **Example_2_L2** и изменим в свойствах проекта общие параметры **...EXAMPLE.000.01** на **...EXAMPLE.000.02**. Заменим **Template** с первого листа рамки ГОСТа на второй. Все это не обязательные требования, а лишь следование требованиям ГОСТа.

Изменение схемы модуля

Изменим частично схему, заменив все **Power Port** на простые **Net Label**, так как все изменения в иерархии вначале мы переда-

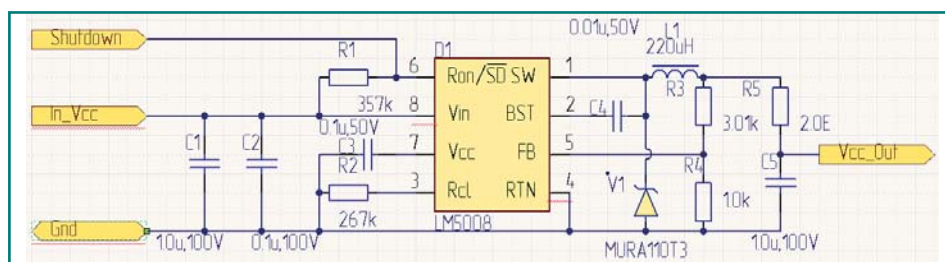


Рис. 22. Пример схемы

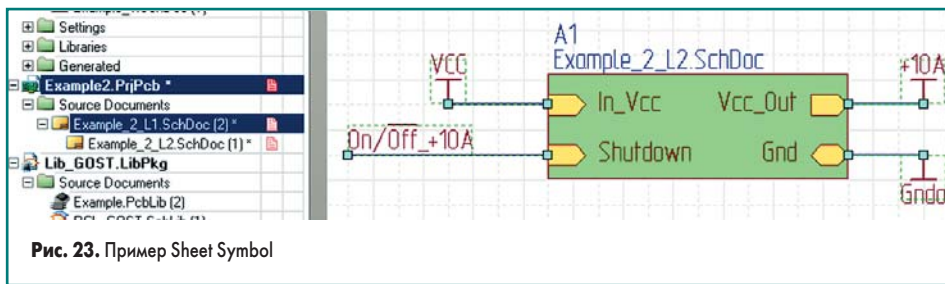


Рис. 23. Пример Sheet Symbol

дим через **Port**. Добавим ко всем входным и выходным сигналам **Port**. После преобразования схема выглядит так, как показано на рис. 22.

Все модули питания схематично будут выглядеть одинаково, за исключением номиналов некоторых элементов схемы. На этом схеме модуля можно считать завершенной.

Создание типовых иерархических схем

Создадим в проекте новый первый лист схемы **Example_2.L1**. Воспользуемся командой **Design>Create Sheet Symbol From Sheet Or HDL**. В открывшемся окне в списке листов проекта появится единственный, отличный от этого лист схемы **Example_2.L2**. Выбираем его и у нас появится на схеме **Sheet Symbol**. Изменяем значение его **Designator** на **A1** Подключаем лист к **Sheet Entry**, как показано на рис. 23 и имеем законченную ссылку на модуль DC/DC для аналогового питания на 10 В. Переход в иерархии производится при нажатой клавише **Ctrl** на изображении **Sheet Symbol**, обратно тоже, но по любому из **Port**. При этом оба документа должны быть открыты. После этого проделайте операцию **Project\Compile PCB Project**, и изображение в панели **Project** для схем проекта будет иметь вложенный характер.

Однако это самый простой, правда, и наиболее распространенный способ применения иерархических схем. Перейдем ко второму слу-

чаю — блоку питания на 3,3 В. Он ничем не отличается от первого, за исключением номиналов резисторов **R3** и **R4**. Ради этого нет смысла полностью перерисовывать схему. Осуществим следующие действия (**AR0112 Multi-Channel Design Concepts**).

1. В уже имеющемся **Sheet Symbol** введем следующие параметры: **V_Out=10V**; **R3=3.01k**; **R4=1.0k**, означающие, что для уровня 10 В номиналы резисторов **R3** и **R4** имеют указанные значения. Сделаем видимым как само имя параметров, так и его значение. Отредактируем **Sheet Symbol**.
2. Скопируем **Sheet Symbol**, изменим его параметры для условия формирования выходного напряжения 3,3 В и добавим соответствующую обвязку.
3. Перейдем на нижний уровень и в параметрах **ValueSCH** для резисторов **R3** и **R4** введем значения, соответственно, **=R3** и **=R4** и сделаем их видимыми, а параметр **Comment** невидимым. После чего следует провести компиляцию проекта, и значения из верхней схемы иерархии передадутся в нижнюю. Параметр из **Sheet Symbol** передается напряжением в указанный параметр конкретного компонента модуля, но через него в **Comment** нет. Будем надеяться, что в ближайших релизах пакета это будет устранено, что принципиально не меняет сущности этого подхода. Переключаясь между модулями **A1**

и **A2** листа **Example_2.L2**, вы наглядно видите изменение номинала резистора **R3**, как и было задано.

Перейдем к введению в схему четырех идентичных модулей на 5 В. После копирования и введения соответствующего параметра для резисторов **R3** и **R4** нужно указать, что таких модулей у нас 4. Для этого в его **Designator** делаем запись **Repeat(A,3,6)**, где

- надпись **Repeat** означает повторяющийся модуль;
- в скобках первый символ «**A**» — это и есть значение **Designator** для модуля;
- 3,6 — первый и последний номер в повторяющейся серии.

Таким образом мы создадим четыре идентичных модуля с позиционными номерами **A3**, **A4**, **A5**, **A6** (рис. 24). Заметим, что линии связи, подводимые к **Sheet Entry** с именами **In_Vcc** (входной уровень питания) и **GND** («земля») являются общими, а **Shutdown** (включение питания) и **Vcc_Out** (выходной уровень питания) в принципе могут отличаться. На последнем и остановимся. Для этих сигналов в **Sheet Entry** также нужно вписать **Repeat(...)**. Соответственно скорректируем схему: для шинных сигналов в квадратных скобках указываем начальное и конечное значение сигнала в шине. Так как мы сами формируем имена связей и одному **Sheet Entry** в нашей схеме соответствуют разные связи, галочку **Project\Project Option:Option-Allow Sheet Enter to Name Net** следует снять.

Мы сознательно остановимся на незавершенной схеме первого листа, так как принцип иерархии она все равно отображает. В общем случае по завершении нужно сделать операцию **Annotate** и **Compile**. При операции **Annotate** вы можете установить порядок листов при присвоении позиционных номеров, но лучше сделать это при помощи операции **Tool\Number Sheets**. При **Compile** у нас создались сообщения об ошибках **Floating Power Object**. Мы их игнорируем, это связано с незавершенностью схемы.

Связь иерархических проектов (AR0123 Connectivity and Multi-Sheet Design)

Итак, мы видим, что работа с иерархическими проектами удобна не только при создании схем, но и далее, при трассировке проекта в целом и блоков в отдельности.

Работа в редакторе PCB

Создаем новую заготовку PCB и передаем данные из схемы в PCB. Для этого используем команды **Design\Update PCB...** при актив-

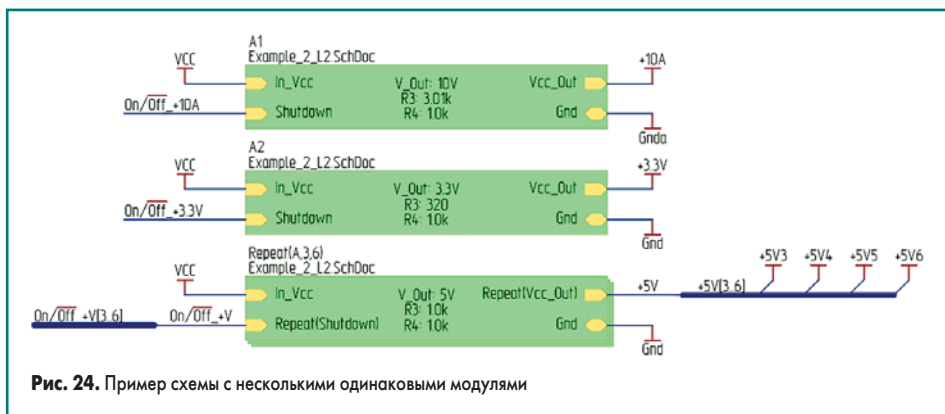


Рис. 24. Пример схемы с несколькими одинаковыми модулями

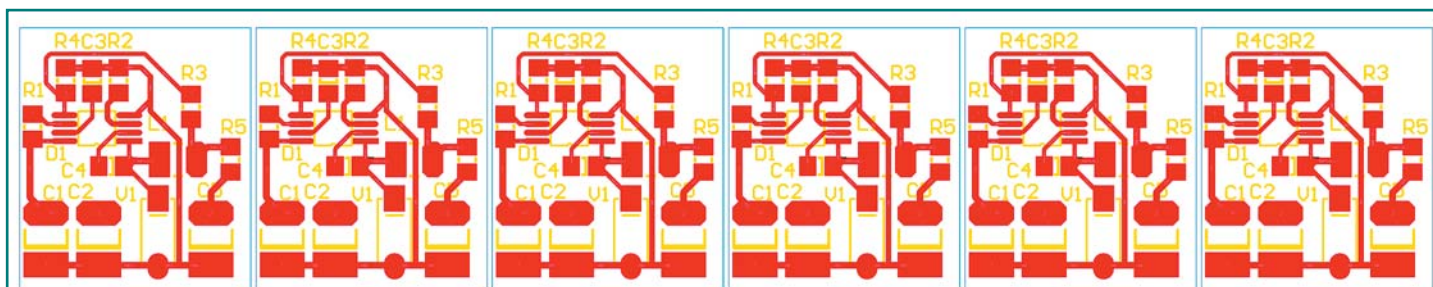


Рис. 25. Результат трассировки 6 идентичных модулей

ной схеме или **Design\Import Change...** При постоянном внесении изменений в схему и РСВ и для полного согласования компонентов схемы и РСВ следует поддерживать и отслеживать связь, используя команду **Project\Component Links**, доступную при активном документе РСВ.

На РСВ видим 6 **Room**. Берем любую из них и, как делали ранее, расставляем компоненты и разводим связи. Делая эти операции, вы могли воспользоваться и результатами предыдущей разводки в проекте **Example**. Однако сейчас на этом останавливаться не будем. У нас на РСВ **Designator** включает имя **Room**. Это, с точки зрения автора, дает много ненужной информации, а главное — длинные надписи сложно затем разместить на слое **TopSilk**. Переключение отображения между физическими позиционными номерами (с включением дополнительной информации) и логическими (только позиционные номера на соответ-

ствующем листе схемы модуля) производится командой **Design\Board Option**.

Перемещаем одну из **Room** на поле, где и производим расстановку элементов и создание топологии в **Room**. С остальными поступаем проще, а именно:

- выбираем команду **Design\Room\Copy Room Format**, мышкой отмечаем **Room**, формат которого берется за основу, затем один из тех **Room**, куда нужно перенести формат;
- устанавливаем все галочки в параметрах для копирования формата и все **Room** (далее вы сможете переносить только формат расположения элементов или разводку, указать конкретные **Room** для переноса формата).

После выполнения команды все **Room** приобрели одинаковый вид. Однако из-за возможного большего размера нашей **Room**, может произойти наложение **Room** друг на друга. Чтобы этого не произошло, необходимо побеспокоиться об этом заранее и разнести **Room**

вручную. Однако и сейчас это можно сделать в автоматическом режиме:

- выделите все **Room**;
- используя команду **Design\Room\Arrange Room**, расставьте выделенные **Room** в выбранной вами последовательности.

При этом перенесутся только те объекты, которые находятся полностью внутри **Room**, а находящиеся за пределами (такое может быть, если **Room** накладывались друг на друга) останутся на месте. В последнем случае операцию **...\Copy Room Format** следует повторить, а остатки линий удалить.

Эти функции существенно сокращают процесс трассировки сложных плат с идентичными блоками. В результате у нас получилась трассировка 6 идентичных блоков (рис. 25). Нам остается вручную произвести трассировку только тех линий связи, которые связывают между собой **Room**.

Окончание следует

Окончание. Начало в № 5'2006

Система проектирования Altium Designer 6

Владимир Пранович

Pranovich@bsu.by

Подходы и примеры реализаций типовых схемных и топологических решений

Создание классов цепей

Классы цепей можно создать:

- непосредственно для всего проекта;
- на схеме с помощью директив (TR0111 Schematic Editor and Object Reference);
- с помощью средства **Board Definition & Rules** PCBDOC редактора (TR0104 Altium Designer Panels Reference).

Автоматическое создание классов всего проекта предусмотрено для всех цепей, принадлежащих:

- одному **BUS** (цепей с одноименными названиями и переменной цифровой частью);
- одному компоненту;
- группе компонентов с одинаковым параметром **ClassName**;
- всем **NET** с одинаковым параметром **ClassName** (данный параметр, относящийся к **BUS**, считается параметром для всех **NET**, принадлежащих этой **BUS**);
- одному листу схемы проекта.

Большой выбор автоматического создания классов существенно облегчает способ их определения и для малых проектов достаточно ограничиться этим набором (рис. 26).

Однако не следует злоупотреблять установкой везде «галок», так как это приведет к огромному множеству классов, с которыми вы с трудом разберетесь. В наших примерах мы вообще не будем прибегать к автоматическому созданию правил.

Задание классов цепей непосредственно в схеме можно сделать с помощью директив **Place\Directive\Net Classes**. При этом определяется имя класса и в параметрах задаются свойства цепей, принадлежащих данному классу. Как правило, подобный способ задания цепей можно рекомендовать, когда проект ведется разными группами разработчиков для схемного редактора и непосредственно работающих с трассированием в PCBDOC. При этом именно разработчик схемы определяет наиболее важные группы цепей и их параметры. В этом случае и имя класса, и его параметры можно сделать видимыми на схеме. Однако они обычно громоздки и не информативны при отображении на твердой копии схемы. Типовая ошибка пользователей, переходящих на данный пакет, состоит в том, что они сразу хотят и начинают прописывать классы прямо на схеме. В примерах, приведенных далее, мы избежим и этого, так как задавать классы и их параметры таким способом достаточно утомительное и (в начале изучения пакета и проектирования первых проектов) бесполезное дело.

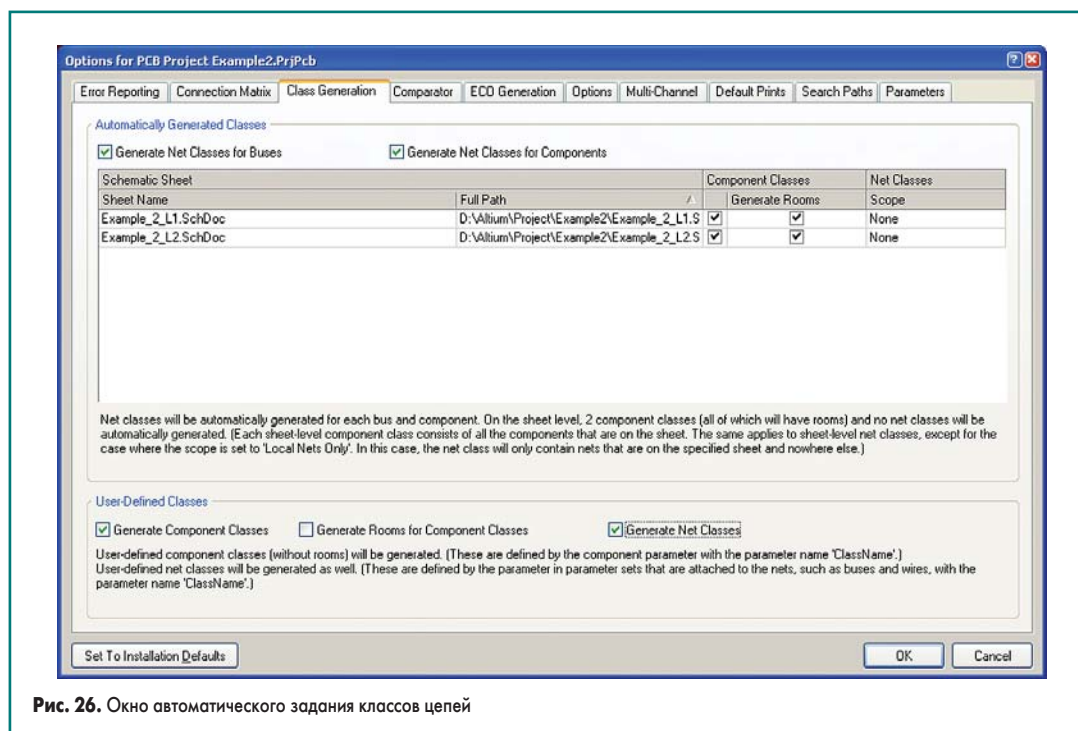


Рис. 26. Окно автоматического задания классов цепей

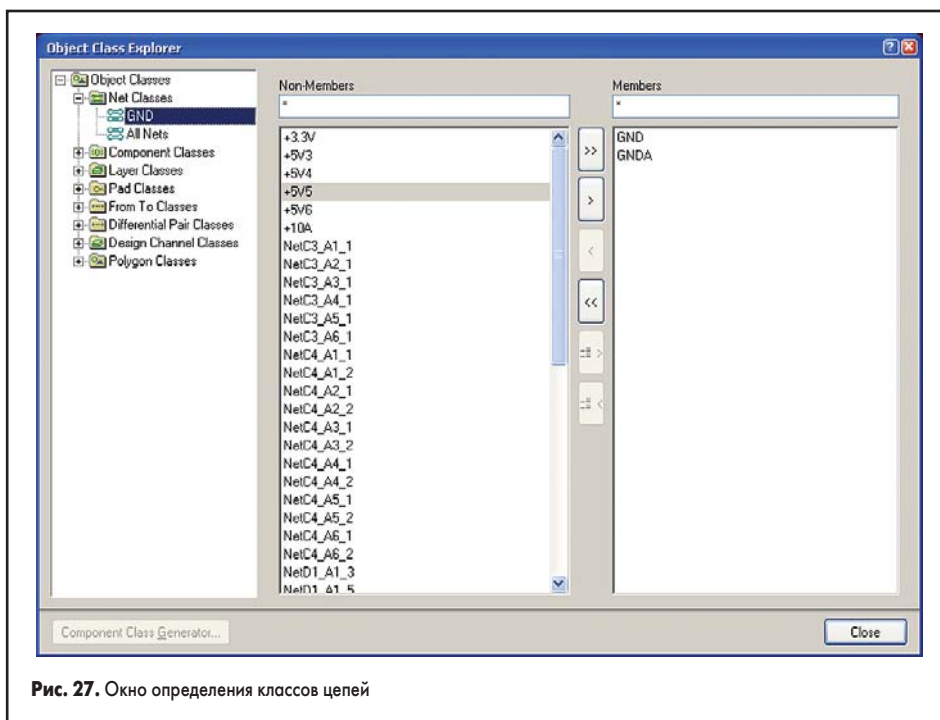


Рис. 27. Окно определения классов цепей

Итак. Определим классы цепей непосредственно в PCBDOC редакторе командой **Design/Classes**. Для нашей схемы введем 2 класса цепей, а именно **GND** и **Power**. Для этого откроем ветку **NetClasses** дерева **ObjectClasses** и щелчком правой кнопки мышки выберем из меню **AddClasses**. При этом у нас сразу добавится новый класс с именем **NewClass**. Аналогично предыдущему, находясь на **NewClass**, выберем из меню **RenameClasses** и введем имя нашего нового класса **GND**. При выделенном классе **GND** отметим в окне **NonMember NET** с именами **GNDA** и **GND** и перенесем их с помощью кнопок в окно **Member**. Таким образом, мы определили класс **GND** и его наполнение и окно **Object class explorer** (рис. 27).

Аналогично добавим класс **Power** и его наполнение. Пользователи легко в дальнейшем определятся и с другими типами классов цепей.

Создание правил для цепей и классов

В примере (см. ТвЭП № 5–6 `2006) при проектировании платы мы задавали пределы для ширины дорожек и зазоров общие, для всех случаев. Теперь попробуем задать разные правила для различных классов цепей, а также другие типовые и наиболее часто употребляемые правила при проектировании печатных плат. В данной статье мы не будем останавливаться на специфических и редко используемых правилах. Не будем также подробно комментировать, почему выбран тот или иной параметр для правила и его значение.

Вызов панели задания правил производится командой **Design Rules**.

Правила Electrical

Рассмотрим одно из самых употребительных правил **Electrical/Clearance**, определяющее минимальные зазоры между электрическими объектами (дорожками, переходными отверстиями и другими объектами), задающими проводящий рисунок платы. Сейчас у нас прописано единственное правило, установленное нами ранее и действующее на все цепи. Для рассмотрения типовых подходов при создании

новых правил, добавим (пусть и не совсем подходящих нашему примеру) 2 новых:

1. Зазор между цепями класса **Power** и другими объектами на слое **TOP** установим равным не менее 0,4 мм.
2. Зазор между цепями класса **GND**, принадлежащими **PAD** прямоугольного типа и объектами типа **VIA** или **PAD**, установим равным не менее 0,3 мм.

Аналогично добавлению нового класса выделим новое правило и нажатием правой кнопки мыши выберем **NewRule**. При этом добавится новое правило **Clearance_1**. Выде-

лим его и переименуем в **Power_0,4mm**. Данное правило имеет тип **BINARY** и определяет правила между двумя типами объектов. В колонке **Where the first...** отмечаем **Net Class** и в выпадающем параметре выбираем наш класс **Power**. В колонке **Where the second...** отмечаем **Lays** и в выпадающем списке параметров выбираем слой **Top Layer**. Устанавливаем требуемый зазор, равный 0,4 мм. Окно правил представлено на рис. 28.

Таким образом, достаточно быстро задать все простые правила. Теперь перейдем к более сложному правилу для второго случая.

Добавим еще одно правило с именем **GND_1_0,3mm**. Однако нам не хватит типовых наборов (первых пяти) в колонках **Where the ...**, поэтому воспользуемся **Advanced (Query)**. Для колонки **Where the first...** дополнительно нажмем кнопку **Query Helper**. В появившемся окне **PCB Function** выбираем **Object Type checks** и его свойство **IsNet**. После чего надпись **IsNet** перемещается в верхнюю часть окна. На панели операндов нажимаем знак «=» и сами дописываем имя цепи — **GND**. Первая часть выражения написана. Теперь на панели операндов нажимаем **And**, а в окне **PCB Function** выбираем **Object Type checks** и его свойство **IsPad**. В верхней части получаем выражение **IsNet=GND And IsPad**. Внизу нажимаем кнопку **Check Syntax**. Если появляется надпись **Expression is o'key**, мы можем принять правило. Так строятся очень сложные правила, и их следует всегда проверять на правильность применения. Для колонки **Where the second...** с целью разнообразия нажмем кнопку **Query Builder**. В левом выпадающем меню выберем условие **Object Kind Is** в правом **PAD** и в правом окошке по-

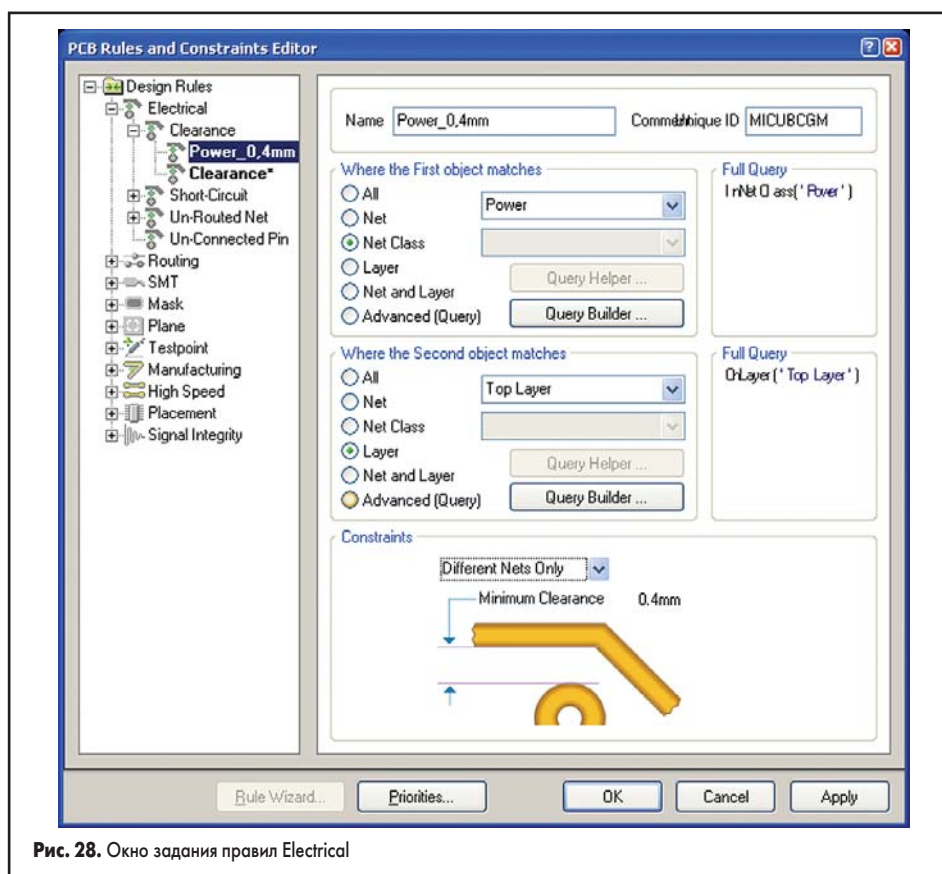


Рис. 28. Окно задания правил Electrical

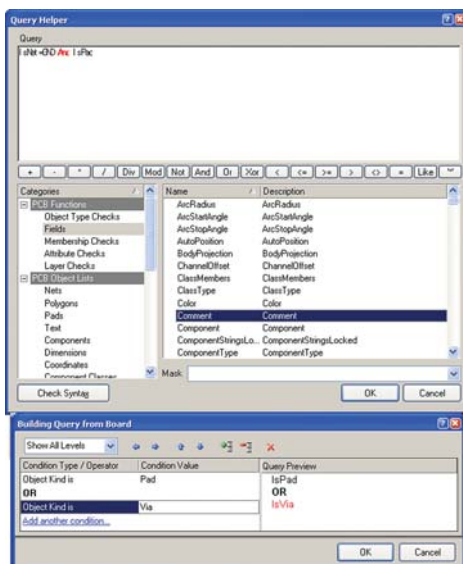


Рис. 29. Окно задания правил

лучим надпись **IsPad**. Выберем в первой колонке надпись **Add another condition**, затем **Object Kind Is** и надпись **IsVia**. Ставим оператор **OR**. В итоге формируется выражение **IsPad OR IsVia**. Здесь строить выражение проще, но меньше вариантов. Вид обоих окон представлен далее. Наконец зададим, как и для предыдущего класса, требуемый зазор и будем считать, что правила **Clearance** определены для нашего проекта (рис. 29).

Правила **Short Circuit** не требуют пояснений. Отметим, что может использоваться для одной физической цепи, которая на схеме может быть обозначена несколькими **NET**, а замкнута только на **PCB**. Также не требуют особых пояснений правила **UnRoutedNet** и **UnConnectedPin**.

Правила Routing

Основное правило **Width**. Правила задания свойств такие же, как и выше, за исключением того, что это правило имеет тип **UNARY**. Здесь пользователь может указать ширину в зависимости от слоя и т. п. Построение правил аналогичное. Единственно отличие — это

правило имеет не одно, а три значения: минимальное, максимальное и установленное по умолчанию, что эффективно используется при интерактивной разводке.

Routing Topology, Priority, Layers, Corners определяют типовой набор правил, характерный и для других аналогичных пакетов, и определяют тип стратегии разводки, приоритет последовательности разводки, направления и стоимости разводки в слоях, вид округления «углов» дорожек при трассировке. Ряд правил **Fanout Control** определяет порядок и форму подвода дорожек к микросхемам для поверхностного монтажа.

Правило **Routing Vias** аналогично **Width** и так же активно используется при интерактивной разводке.

Отдельно в этом ряду стоит правило **Diff-PairsRouting** для разводки дифференциальных пар. Однако оно не должно вызывать больших затруднений.

Специальная группа правил **SMT** формирует порядок подвода дорожек к **SMD**-элементам и связана с технологическими требованиями монтажа таких элементов: как правило, при автоматической пайке для обеспечения требуемого термобарьера; исключения «наплыва» припоя при «остром» угле подключения дорожки (близком положении угла поворота дорожки) к **SMD**-площадке или близком положении угла поворота дорожки.

Правила Mask

Правило **Solder Mask Expansion** определяет зазор между защитной маской и размерами **PAD**. По умолчанию зазор установлен в 0,1 мм. Пользователь вправе его увеличить и уменьшить, более того, сделать отрицательным. Последний способ можно использовать, когда, например, **PAD** увеличенного размера используется для крепления транзисторов с целью повышения теплоотвода, при этом теплоотвод обеспечивается и с учетом закрытия части **PAD** маской. Правило **Paste Mask Expansion** определяет зазор при изготовлении трафарета нанесения пасты.

Правила Plane

Правила задают способ и параметры (включая термобарьеры) подключения слоев **Plane** и полигонов к переходным отверстиям и **PAD**. Рассмотрим только правило **PolygonConnect**. Правило задает способ подключения **PAD** к полигону:

- **Relief Connect** — с термобарьером, при этом определяется число и ширина **Relief**, а также угол подвода к **PAD**;
- **DirectConnect** — без термобарьера;
- **NoConnect** — запрет подключения.

Обычно эти правила различны для разных полигонов и прописываются аналогично правилу **Clearance**. В нашем случае ограничимся созданием одного правила **PolygonConnectPadMultiLayer**, разрешающего подключению к полигону всех **PAD** для штыревых элементов, как показано на рис. 30.

Правила Testpoint, Manufacturing, High Speed, Signal Integrity

Из большой группы этих правил рассмотрим лишь несколько.

Правило **HoleSize**. Определим для нашего проекта два правила. Одно для всех цепей с минимальным (0,5 мм) и максимальным (1 мм) размером **HOLE**, и второе — для силовых цепей, как показано на рис. 31, с увеличенными значениями **HOLE** (для обеспечения требований максимального тока для силовых линий).

Правило **ViaUndoSMD**. Установим галочку **Allow vias undo smd Pads**. Предположим, что наша плата будет только ручной сборки, и главным требованием является плотность сборки и минимизация числа используемых слоев.

Правило **ComponentClearance**. В данном правиле оставим стандартные зазоры и установим **CheckMode=FullCheck**. Связано это с тем, что при создании посадочных мест мы не делали ни **ComponentBody**, ни контур **Courtyard** для допуска посадочных мест. В будущем рекомендуем сразу формировать данные изображения, тем более что в команде **Tools/IPC Footprint Wizard** появилась фун-

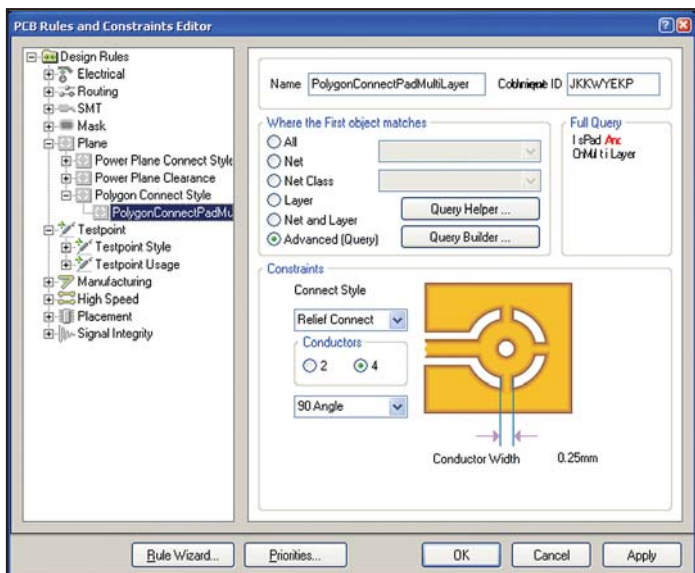


Рис. 30. Окно задания правил Plane

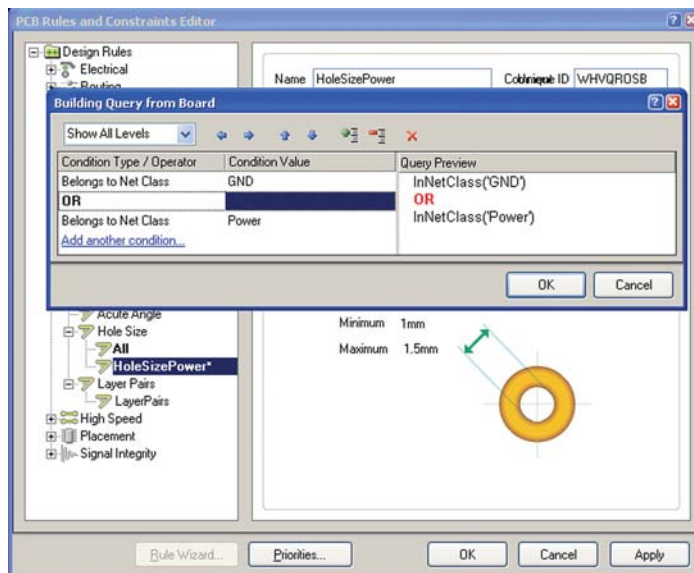


Рис. 31. Окно задания правил HoleSize

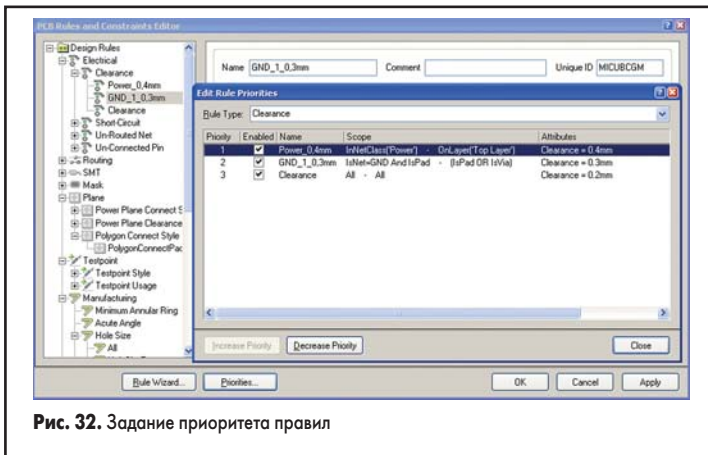


Рис. 32. Задание приоритета правил

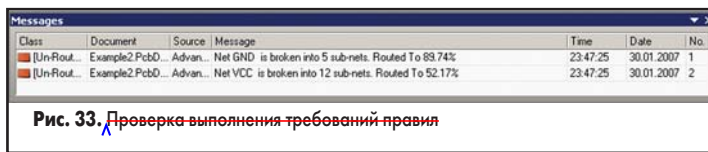


Рис. 33. Проверка выполнения требований правил

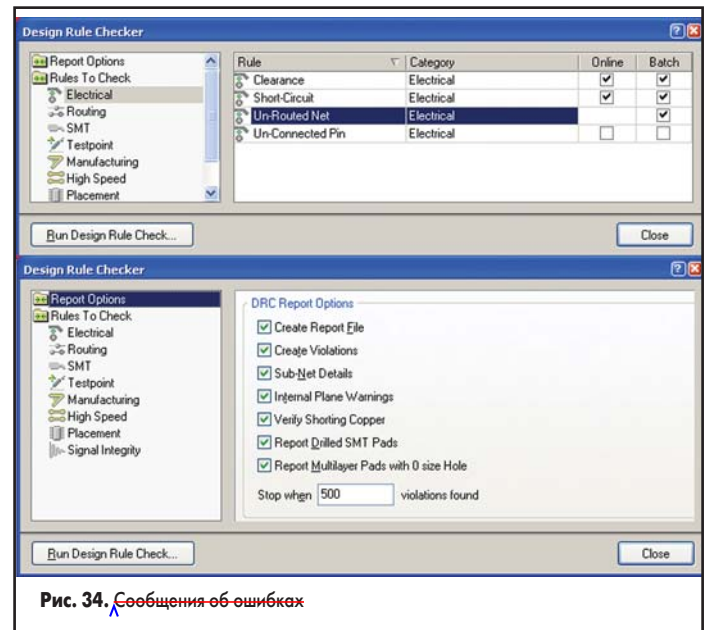


Рис. 34. Сообщения об ошибках

кция автоматического формирования данных изображений для множества типов корпусов. В этом случае можно использовать и другие значения параметра **CheckMode**. При этом проверка выполнения этого параметра будет доступна не только при запуске программы проверки, но и в интерактивном размещении, и произойдет значительно быстрее.

Параметр **Height**. Его следует вводить и определять для специально созданных или имеющихся **ROOM**.

Задание приоритета правил и проверка их выполнения

На этом будем считать завершенным создание правил для нашего примера и перейдем к заданию приоритетов правил. Задать приоритет правил можно непосредственно в окне **PCB Rule and...** нажатием кнопки **Priorities** для определенного типа правил. Далее представлена установка таких приоритетов для параметра **Clearance** в нашем примере. Аналогично проверяем и устанавливаем порядок приоритетов для всех правил одного типа (рис. 32).

Многие правила одинаковы для разных проектов. И более того, у пользователей образуется целый свод любимых правил. Нет необходимости для каждого проекта так долго их вводить. Щелчок правой кнопкой мышки в левой части окна **PCB Rule and...** вызывает меню, позволяющее сделать экспорт или импорт выбранных правил в отдельный файл с расширением *.rul. Можно в этом файле хранить наиболее часто используемые правила и затем импортировать их в новые проекты.

Так как мы ввели не только много новых правил, но и изменили значения параметров существующих, следует проверить проект на соблюдение требований новых правил. Делаем это с помощью команды **Tolls/Design Rule Check**, установив в ней проверку нужных для нас параметров. Параметры, отмеченные в колонке **OnLine**, будут всегда проверяться при интерактивной разводке, а отмеченные в колонке **Batch** — только при запуске указанной программой. Здесь же, в окне **Report Op-**

tion, можно указать опции формирования Report-файла (рис. 33).

В результате проверки у нас есть сообщения об ошибках. Откроем панель **System/Messages** (рис. 34).

Ни одно из новых правил не нарушено. Две ошибки свидетельствуют о незавершенности нашего проекта, когда мы делали иерархический проект. И теперь необходимо завершить разводку **NET** (а именно **GND** и **VCC**), присутствующих в нескольких иерархических блоках.

Создание полигонов (TR0112 PCB Editor and Object Reference)

Завершив разводку нашего проекта. Одновременно введем в наш пример несколько полигонов и применим разные требования к ним.

1. Первый связан с **NET=GND**. Главное требование — обеспечить экранизацию различных участков.
2. Второй связан с **NET=GND**. Главное требование — обеспечить максимум заливки и максимально широкие области при подводке к **PAD** с большими импульсными и постоянными значениями тока.
3. Третий связан с **NET=VCC**. Главное требование — обеспечить только максимально широкие области при подводке к **PAD** элементов, потребляющих большую мощность. Для первого случая включим настройки, как показано на рис. 35.

При прорисовке сложных контуров полигона не всегда сразу удается правильно расположить его углы. Для доступа к изменению сложной формы полигона, выдели его, следует затем нажать правую кнопку мышки и в контекстном меню выбрать **Polygon Action/Move vertex**. После этого в углах полигона (как на рис. 35) появятся их метки, которые можно переместить в требуемое положение.

Мы указали область заливки сплошной. Однако все **PAD** цепи **GNDA** оказались присоединенными к полигону. Для аналоговой «земли» желательно избежать такого соединения, а полигон подключить к данной цепи

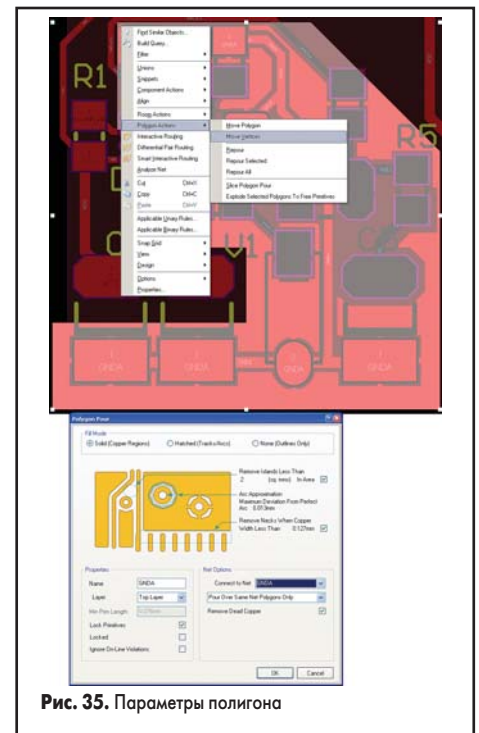


Рис. 35. Параметры полигона

только в одной точке. Более того, данный полигон следует соединить затем и с полигоном **GND** в этой же точке. Для реализации этого придется вернуться к правилам и запретить подключение **PAD** к полигону **GNDA** одним из рассмотренных выше способов. Для подключения полигона **GNDA** к **NET** и соединения его с другим полигоном **GND** создадим новый библиотечный элемент, например, из двух **PIN**, указав его тип **Tie Net** и соединив на схеме с его помощью цепи **GNDA** и **GND**, как показано на рис. 36. При создании посадочного места для компонента **TIE** его **PAD** делаем с перекрытием или иной формы, обеспечивающей физическое замыкание между контактами.

После внесения всех изменений наш полигон будет выглядеть так. Все соединения с **NET GNDA** произведены с помощью дорожек. Сам полигон занимает все свободное место в соответствии с установленными зазорами и под-

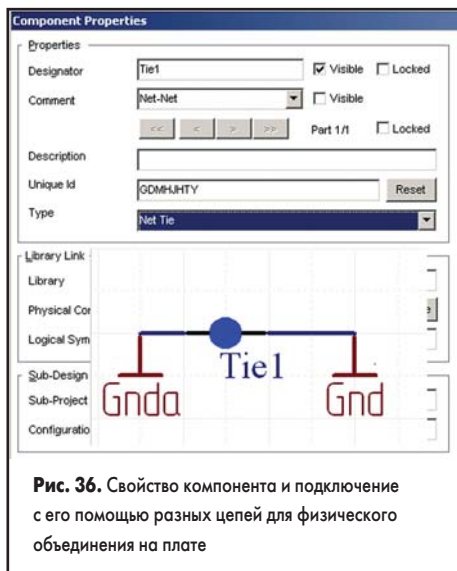


Рис. 36. Свойство компонента и подключение с его помощью разных цепей для физического объединения на плате

ключен только в одной точке (VIA возле компонента TIE1). Объединение NET GND и GND произведено типологически, через PAD компонента TIE1. В этом случае любое пересечение или несоблюдение зазоров при разводке NET GND и GND (кроме места компонента TIE1) вызовет сообщение о нарушении правил трассировки.

Создание полигона для NET GND наиболее просто. Аналогично предыдущему делаем прямоугольный полигон на всю область нашей платы. Полигон GND на рис. 38 выделен более ярким цветом. Так как он делался после полигона GND, область заливки обтекает GND этого полигона.

Однако теперь нет места для задания третьего полигона VCC. Далее будет приведено ре-

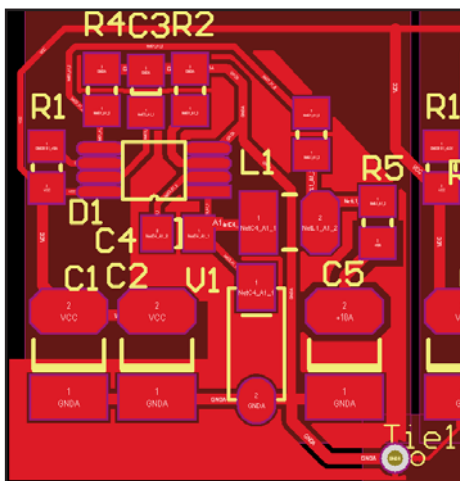


Рис. 37. Вид полигона с подключением в одной точке

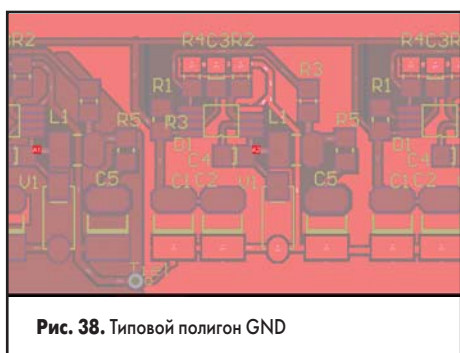


Рис. 38. Типовой полигон GND

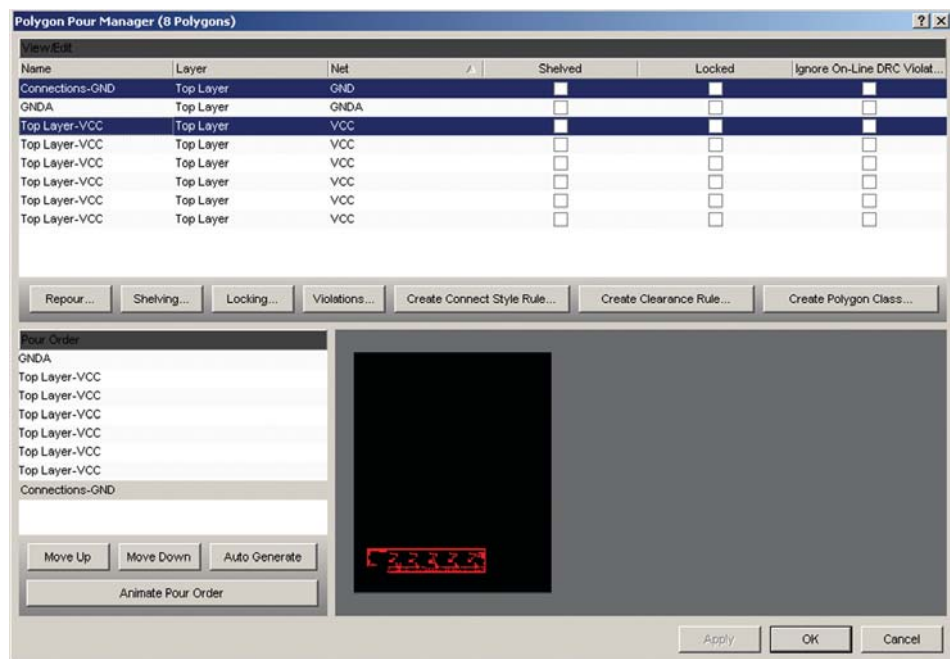


Рис. 39. Tools/Polygon Pour/Polygon Manager и итоговый вид полигонов

шение, и не только для этого случая. Как правило, полигоны бывают большие и требуют много времени на их постоянную перезаливку. Для того чтобы полигон не мешал текущей разводке или внесению изменения, мы можем временно скрыть один или все полигоны. Для этого воспользуемся командой Tools/Polygon Pour/Shelve.... Вид полигонов станет неотображаемым, и мы можем приступить к формированию полигона VCC. Так как нам нужно только увеличить области подвода шин питания, лучше сделать это не одним полигоном, а группой идентичных. После проведения все манипуляций мы получим удовлетворяющую нас заливку полигона VCC. Теперь нам важна последовательность заливки полигонов: сначала — GND, как наиболее важный и ответственный, затем группу полигонов VCC, для обеспечения его первичной заливки, и, наконец, GND — для заливки оставшейся свободной поверхности. Можно, конечно, скрыть все, и затем по очереди в необходимой последовательности открывать и снова заливать. Это просто для элементарных проектов, где число полигонов ограничено. Для проектов с большим числом разнообразных типов полигонов лучше воспользоваться Tools/Polygon Pour/Polygon Manager (рис. 39), где можно указать как порядок заливки полигонов, так и оперативно внести изменения в некоторые параметры. Там же показан итоговый результат проекта со всеми полигонами.

Применение SWAP в PCBDOC (AP0138 Pin and Part Swapping with Dynamic Net Assignment)

В связи с появлением одноконтурных компонентов логических элементов и широким распространением многослойных плат применение данной функции резко сократилось. По крайней мере, в более чем 2-летней практике автора эта функция ни разу не была вос-

требована. Однако большинство пользователей, переходящих со старых версий других САПР, пользовались такой функцией и по традиции считают ее необходимой. В последней версии Altium Designer 6 такая возможность появилась, и мы вкратце остановимся на ней.

Добавим в нашу схему блок, показанный на рис. 40. В блоке присутствуют компоненты с рядом эквивалентных PIN и PART.

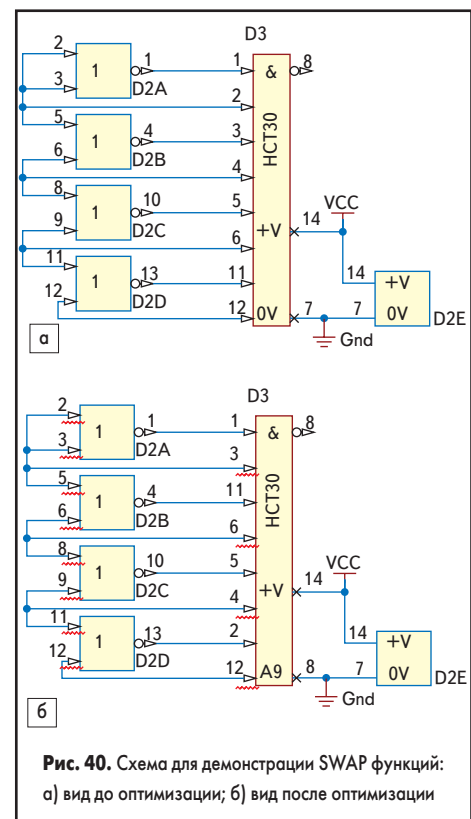


Рис. 40. Схема для демонстрации SWAP функций: а) вид до оптимизации; б) вид после оптимизации

Для задания эквивалентности PIN и PART используется команда Tools/Configure Pin Swapping. При этом вызывается таблица компонентов проекта, где можно выбрать компонент и затем в соответствующей таблице задать эквивалентность PIN и PART (рис. 41).

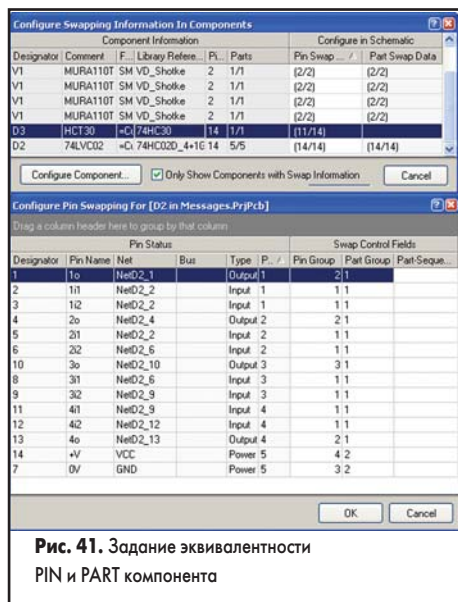


Рис. 41. Задание эквивалентности PIN и PART компонента

После передачи изменений в PCBDOC документ и расположения элементов с помощью команды **Tools/ Pin/Part Swapping/Configure** разрешаем указанным компонентам возможность взаимно замещения эквивалентных PIN и PART. Затем одной из команд данной группы применяем команду на применение оптимального или ручного замещения. В результате оптимизации произведена замена 9 из 11 возможных замен, а также выводится и другая справочная информация (рис. 42).

Теперь введем изменения, сделанные в PCB, в схему с помощью команды **Design/Update Schematic...**

Базы данных (AP0134 Linking Existing Components to Your Company Database)

В заключение затронем еще одно качество САПР, которое пока редко применяется пользователями, однако оно дает прекрасную возможность для совместной работы над одним проектом, снижения ошибок при проектировании и быстром получении и модификации сопроводительной информации — это подключение баз данных. База может быть подключена:

- к проекту для автоматического переноса параметров компонентов из базы в схему;
- инсталлирована как библиотека, для выбора самих компонентов прямо из базы.

Как правило, большинство компонентов при проектировании схемы ставится в нее без учета номиналов, вариантов посадочных мест и других параметров, которые могут быть изменены в процессе проектирования. При этом изменение одного из параметров влечет за собой и изменение других (например, изменение номинала или завода изготовителя компонента — изменение наименования). Постоянное отслеживание соответствия всех параметров компонентов занимает много времени и не повышает производительности. Подключить базу данных можно и из примера, входящего в пакет САПР. Однако мы создадим свою базу: это достаточно просто и понятно.

Произведем следующую последовательность действий:

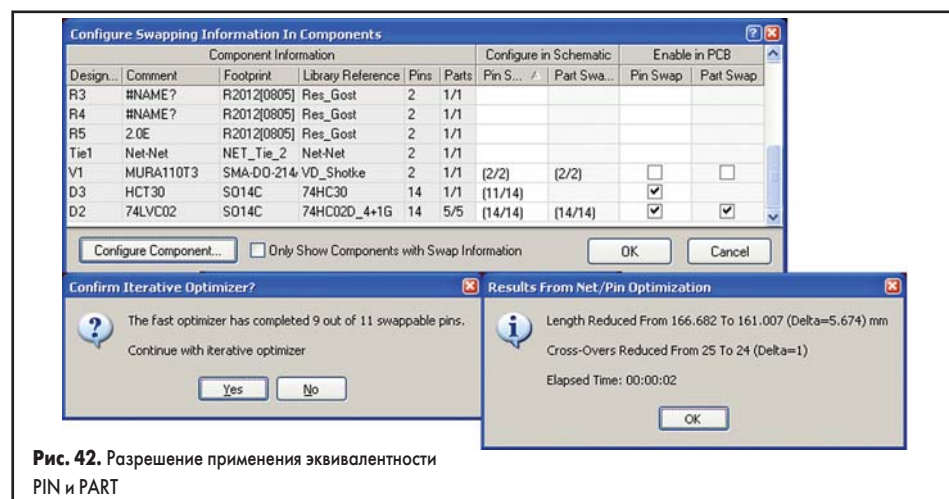


Рис. 42. Разрешение применения эквивалентности PIN и PART

1. Из схемного редактора командой **Tools/Parameter Manager**, как уже делали ранее, откроем таблицу параметров компонентов.
2. Выделим столбцы существующих параметров и скопируем их в буфер.
3. Откроем программу EXCEL и вставим значения из буфера.
4. Отсортируем строки, удалим дубликаты строк и столбцы с параметрами, которые мы не будем использовать.
5. Добавим первую строку и введем наименования столбцов.
6. Расположим столбцы в удобной для нас последовательности.
7. Добавим первый столбец в пустой столбец с именем, например Link.
8. В столбце Link введем уникальный номер для каждой строки, отражающий свойства компонента. Желательно ограничить его длину, так как для сверки с базой необходимы версии твердых копий схем с видимым значением этого параметра. В частности, в примере мы сформируем его название из типа корпуса и обозначения основного параметра.
9. По желанию можем добавить и заполнить и другие столбцы с информацией о поставщике, стоимости изделий, ссылки на DataSheet и так далее.
10. Сверяем таблицу с исходными данными, находим и устраняем несоответствие в параметрах.

Итак база готова, ее вид показан на рис. 43. Сохраняем его в наш проект, например, с именем Example2.

Link	DataSheet	Наименование	ValueSCH	MarkPCB	Поставщик
C4532/105		Конд. TDK C4532X7R2A105M	1.0u,100V		
C3216/104		Конд. TDK C3216X7R2A104M	0.1u,100V		
C1206/104		Конд. Kemet C1206C104K5RAC	0.1u,50V		
C1206/103		Конд. Kemet C1206C103K5RAC	0.01u,50V		
C4532/156		Конд. TDK C4532X7R1E156M	15u,100V		
MSOP/5008		М-с. LM5008, NC	LM5008		
SO/LVC20		М-с 74LVC02	74LVC02	*	
SO/HCT30		М-с IN74HCT30D, Интеграл	HCT30	*	
L1014/221		Дроссель TDK SLF10145T-221MR65	220uH		
R1206/3573		Рез. Vishay CRCW12063573F	357k		
R1206/2673		Рез. Vishay CRCW12062673F	267k		
R1206/3011		Рез. Vishay CRCW12063011F	R3		
R1206/1001		Рез. Vishay CRCW12061001F	R4		102
R1206/2R00		Рез. Vishay CRCW12062R00F	2.0E		20
DO214/MURA110		Диод ON Semi MURA110T3	MURA110T3		

Рис. 43. Готовая база

Компоненты в схеме можно связать по любым параметрам, но так как мы хотим наоборот изменять параметры в схеме, введем в ней ко всем элементам параметр **Link** и его значение в соответствии со столбцом **Link** нашей базы. С этого момента и в дальнейшем достаточно отслеживать только этот параметр и его значение в схеме, а остальные вводить или через базу или, не вводя в схему, сразу добавлять их значения в документы при генерации REPORT.

Добавим **Project/Add New To Project** к нашему проекту новый документ **DataBaseLinkDate** и сохраним с именем **Example**. В документе **Example.DBLink** отмечаем необходимые установки и производим следующие действия:

1. Создаем базу в EXCEL, в выпадающем меню **Select Database Type=Microsoft Excel**.
2. Нажимаем кнопку **Browse** и указываем путь к файлу базы данных.
3. Нажимаем кнопку **Connected**.
4. Выбираем параметры для проекта и баз данных, по которым будут связываться остальные параметры. В нашем примере имена параметров совпадают и имеют значение **Link**.
5. Каждому параметру, кроме **Link**, можно поставить атрибут или обновить его в схеме, или добавить в схему, если его нет, и так далее.

База данных готова.

Теперь в схемном редакторе можно использовать команду **Tools/Update from Database**, и при генерации различных REPORT до-

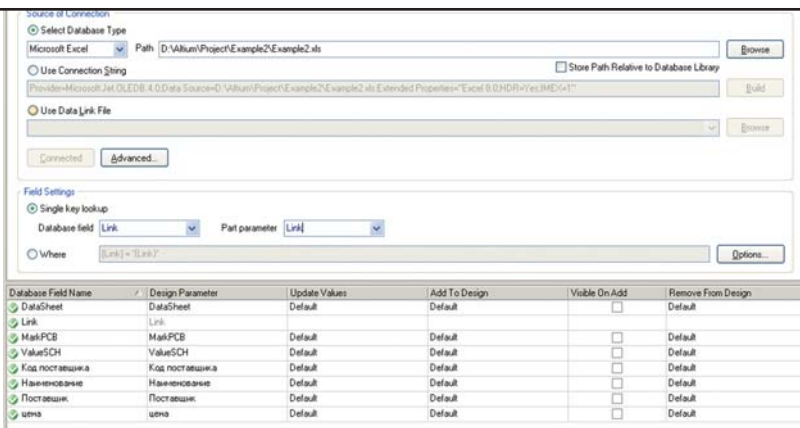


Рис. 44. Настройка базы данных

бавлять дополнительные параметры, которые находятся в базе данных (рис. 44).

Создание базы данных библиотечных элементов

В качестве заготовки базы используем лист EXCEL файла, в котором нужно добавить столбцы:

- Part Number — для заполнения значения имени компонента в библиотеке;
- Footprint Ref — для заполнения значения имени посадочного места данного компонента;
- Library Ref — для автоматического заполнения данного параметра компонента в библиотеке;
- другие столбцы с параметрами по желанию пользователя.

С помощью команды **File/New/Library/Da-taBaseLibrary** создадим новую базу данных, сохраним ее под именем **Example** и подключим, как делали выше, созданный EXCEL файл. Дополнительно, через **Tools/Option**, указываем директорию, где находятся библиотеки, в которых будет проводиться поиск компонентов по названиям из базы данных. Через панель **Library** проинсталлируем новую, только что сделанную нами библиотеку **Example.DbLib**.

Теперь прямо с панели **Library** мы сможем вводить и в схему, и в РСВ новые компоненты, которые мы сами определили в EXCEL файл, изображение и посадочные места для которых существуют хотя бы в одной из проинсталлированных стандартных библиотек.

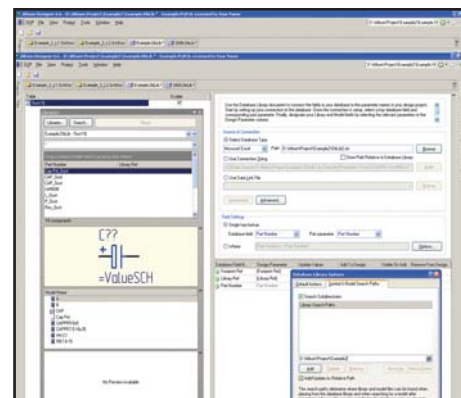


Рис. 45. Настройка базы данных DbLib

Пример заполнения параметров и свойств базы данных и вид панели **Library** при выборе компонента из библиотеки представлен на рис. 45.

Теперь добавление нового библиотечного компонента, не имеющего отдельного изображения на схеме и специфического посадочного места, производится путем записи в базе, даже без открытия пакета САПР.

Заключение

Автор надеется, что его статья позволит снять целый ряд неизбежных вопросов, возникающих при освоении данного пакета САПР, окажет помощь пользователям в быстром переходе от изучения к практической реализации первых собственных проектов и сократит время до получения полностью завершённых работ от разработки до получения печатных плат.