

Altium Designer 6 в примерах

В статье приводится последовательность действий при создании библиотек, разработке схемы, трассировки платы и подготовки выходных файлов и технологических и других документов на конкретных примерах. Автор не претендует на уникальность и полноту предложенных подходов, а просто приводит один из способов по написанию правил, созданию классов, компонентов и других действий, которые были применены в приведенных проектах. Многие способы и приемы в статье рассмотрены поверхностно или вообще не затронуты, так как подробно описаны автором ранее [1].

Владимир Пранович,
К. Т. Н.

pranovich@bsu.by

Пример 1. Аналого-цифровой преобразователь

Настройка приложения и проекта

Сделаем настройку нашего первого проекта так, как он настроен у автора. При этих настройках вы всегда будете видеть на экране именно то, что будет приведено ниже на рисунках, и все предложенные действия будут приводить к одинаковым визуальным результатам. Для этого выполним следующие действия:

1. Установите их на своем компьютере шрифты, удовлетворяющие требованиям ГОСТ. В качестве таких можно использовать шрифты от других пакетов.
2. Командой **File/New/Design Workspace** создадим новый дизайн. По умолчанию создается файл **Workspace1.DsnWrk**. Для удобства командой **File/Save Design Workspace As** сохраним его под именем **Training.DsnWrk** в директорию **...DsnWrk**. Здесь и далее всегда будут указаны относительные пути размещения директорий и файлов. Место расположения вы можете устанавливать самостоятельно.
3. Командой **File\New\Project\PCB Project** создадим новый проект. По умолчанию создается файл **PCB_Project1.PrjPCB**. Командой **File/Save/Project As** сохраним его как **...Project\Training_01\Training_01.PrjPCB**.
4. Для выбора шрифта, используемого в качестве системного, применим команду **«Design/Document Option/Sheet Option/Change System Font»**. В примерах автора будет использоваться шрифт GOST type A, установленный ранее (см. п. 1).

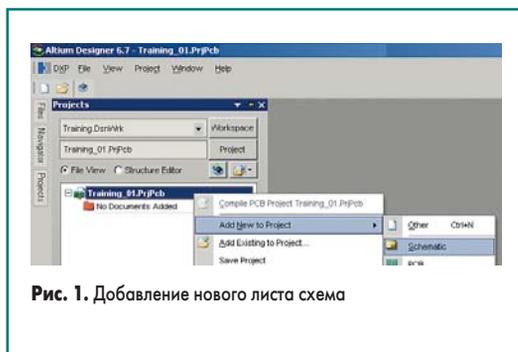


Рис. 1. Добавление нового листа схема

5. Из панели **Project** нажатием правой кнопки «мышки» при выделении нашего проекта вызываем контекстное меню (рис. 1), командой **Add New to project/Schematic** создаем новый лист схемы проекта и командой **File/Save As** сохраняем под именем **Tr01Sh01**, означающем первый лист нашего проекта. Систему именования листов вы можете использовать свою.
6. В примерах для схем будем использовать размер листа A4. В данной статье мы не останавливаемся на вопросе создания шаблонов (**TEMPLATE**). Вы можете создать свои шаблоны или обратиться к автору. Загрузим шаблон первого листа командой **Design/Template/Set Template File Name**, где укажем путь и имя шаблона (для первого листа формата A4 с вертикальным расположением). При первой загрузке всегда устанавливайте опцию **«Add new parameters exist in the template only»**. При загрузке шаблона на листе будут определены размер, параметры и установки из указанного шаблона.

Выбор схемы для примера

Все примеры в данной статье взяты из реальных проектов, разработанных при участии автора, но приводятся без описания принципа их действия и назначения, а также без обоснования выбора того или иного схематического и конструктивного решения.

Схема, выбранная для примера, приведена на рис. 2. Здесь представлена реализация схемы аналого-цифрового преобразователя на основе ADS1255 (<http://focus.ti.com/docs/prod/folders/print/ads1255.html>). Данная схема не имеет самостоятельного применения и используется в составе других устройств. Однако это не мешает использовать ее в качестве примера. Более того, в дальнейшем мы используем ее при построении схемы многоканального АЦП.

Для создания схемы нам потребуется создать три библиотечных элемента: резисторную и конденсаторную сборки, а также компонент ADS1255. Дополнительно будем вести базу данных всех компонентов, создаваемых или используемых в рамках проекта. Ведение базы — это необязательный элемент, и по началу вызовет немало сомнений в необходимости его использования, но для больших проектов (которые ведет не один разработчик) и на стадии оформления сопроводительных документов он су-

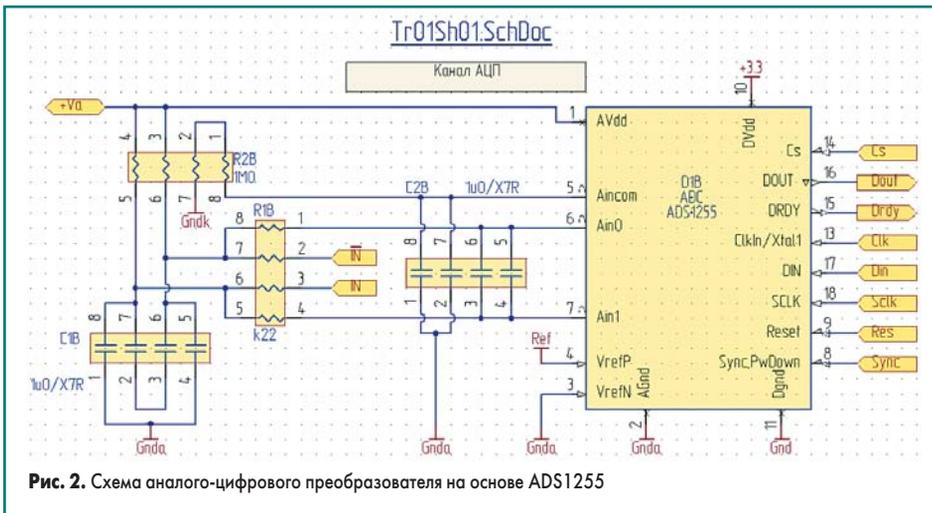


Рис. 2. Схема аналого-цифрового преобразователя на основе ADS1255

щественно облегчает подготовку и оперативное внесение изменений в проект.

Поиск компонентов схемы и пример ведения базы данных

Как правило, все пользователи CAD превосходно владеют программой Excel из пакета Microsoft Office, поэтому предложим пример ведения базы компонентов проекта именно в этом приложении. Тем более что поиск и заполнение ячеек базы может вести и вспомогательный персонал.

1. Создадим в директории ...**Project\Training_01\DOC** базу данных на основе пустого листа Excel (имя файла, например, **DB_TR01.XLS**).
2. Из панели **Project** нажатием правой кнопки «мышки» при выделении нашего проекта вызываем контекстное меню и командой **Add Existing to Project** добавим файл ...**Project\Training_01\DOC\DB_TR01.XLS** к проекту. После этого ссылка на соответствующий файл появится в соответствующем разделе браузера (рис. 3) панели **Project**, и мы сможем открывать этот файл непосредственно из пакета **Altium Designer**. Это

очень удобно, так как не нужно запоминать места и имена файлов, имеющих прямое отношение к проекту, несмотря на то, что они созданы и будут редактироваться не средствами пакета **Altium Designer**.

3. Переименуем лист Excel на **DB_TR01** и введем название столбцов (таблица).

Вы вправе добавить, удалить, переименовать любые столбцы. Обязательно присутствие столбца **L_DB**. На него накладывается еще одно ограничение — в столбце не должно быть совпадающих записей. Именно по записям этого столбца мы будем и добавлять, и модифицировать параметры компонентов проекта.

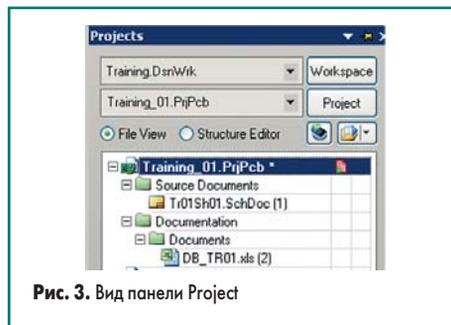


Рис. 3. Вид панели Project

Таблица

Название столбца	Назначение записи в столбце
A — №	Для последовательной нумерации строк и удобства идентификации записей
B — L_DB	Link Data Base. Для связи с проектом (схемой)
C — Item ID	Для указания в базе данных имени компонента схемы
D — Item Footprint	Для указания в базе данных имени посадочного места
E — Modify	Дата ввода или модификации строки
F — PDF	Для указания ссылки на описание компонента
G — Description Pdf	Для внесения краткого описания из PDF на компонент
H — ValueSCH	Для ввода отображаемого на схеме параметра компонента
H — NoteSCH	Для ввода отображаемого на схеме дополнительного параметра компонента
J — Description	Для ввода краткого описания компонента (пояснительная часть записи графы «Наименование» перечня элементов)
K — Supplier	Для ввода имени (ссылки) поставщика компонента (каталожного)
L — Supplier P/N	Для ввода номера компонента по каталогу
M — Vendor	Для ввода имени изготовителя компонента
N — Vendor P/N	Для ввода обозначения компонента
O — ValueBOM	В данной ячейке будут храниться данные для графы «Наименование» перечня элементов. В ячейку вводим формулу: «=СЦЕПИТЬ(J2;»; «N2;»; «M2);», и в ней будет автоматически формироваться запись краткого описания компонента, его обозначение и имя производителя
P — NoteBOM	Для ввода записи, помещаемой в графу «Примечание» перечня элементов
Q — Provider	Для ввода имени (ссылки) компании-продавца компонента
R — Package	Для ввода наименования корпуса посадочного места по PDF
S — Package Description	Для ввода описания корпуса посадочного места по PDF
T — Marking	Для ввода маркировки, нанесенной на корпус

4. Начнем заполнение первой строки листа Excel для идентичных компонентов **C1B** и **C2B** схемы (рис. 2). Для этого:
 - в столбец «№» вводим номер порядковой записи — «001»;
 - в столбец «L_DB» введем запись «C01», в нашем случае это запись для первой модификации конденсатора (буква **C** — обозначение типа элемента, **01** — первый компонент типа конденсатор в базе данных);
 - столбцы «Item ID» и «Item Footprint» оставляем незаполненными, так как на данный момент мы еще не создали компонент и посадочное место;
 - в столбец «Modify» вносим текущую дату;
 - столбцы «PDF» и «Description Pdf» также на данный момент оставляем незаполненными;
 - в столбец «ValueSCH» и «NoteSCH» введем записи, которые будут отражаться на схеме «1u0» — емкость конденсатора и «X7R» — ТКЕ конденсатора (для нашего случая конденсаторы используются в фильтре и данный параметр имеет принципиальное значение).
5. Откроем один из сайтов каталожных поставщиков компонентов, например, www.digi-key.com, сделаем поиск конденсаторных сборок с параметрами Capacitors->Arrays->1.0 uF->X7R и выберем требуемую сборку конденсаторов, доступную на складе с минимальным или необходимым посадочным местом и удовлетворяющую другим существующим требованиям. Копированием из окна поиска (рис. 4) вводим в следующие ячейки значения:
 - в столбец «Description» — «CAP 4-ARRAY 1.0UF 6.3V X7R 1206»;
 - в столбец «Supplier» — «Digi-Key»;
 - в столбец «Supplier P/N» — «490-3446-2-ND»;
 - в столбец «Vendor» — «Murata»;
 - в столбец «Vendor P/N» — «GNM314R70J105MA01L»;
 - в столбец «Package» — «1206 (3216)»;

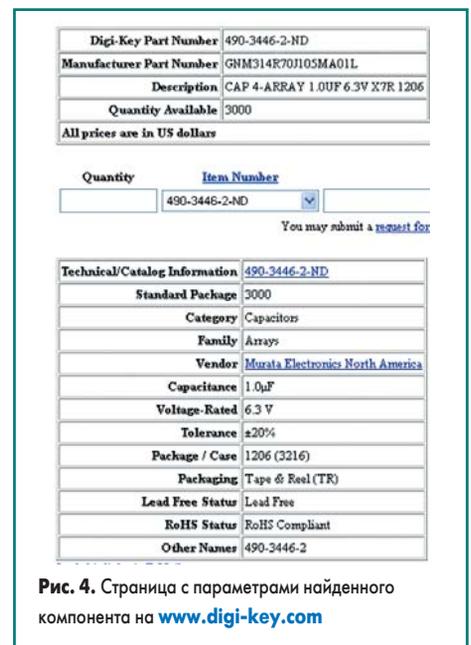


Рис. 4. Страница с параметрами найденного компонента на www.digi-key.com

– в столбце «ValueBOM»

автоматически сформируется запись «CAP 4-ARRAY 1.0UF 6.3V X7R 1206», GNM314R70J105MA01L, Murata».

6. Для данной сборки конденсаторов достаточно описания, приведенного на сайте www.digi-key.com. Поиском по этому сайту находим описание данной сборки конденсаторов на странице 1537 каталога, скачиваем описание и сохраняем в архив описаний под именем «ARRAY TYPE MULTILAYER CERAMIC Murata P1537». Отметим, что лучше описание находить непосредственно на сайте производителя, однако в данном случае ограничимся этим. Открываем документ и заполняем следующие столбцы:

– в столбце «PDF» вносим ссылку на найденный документ ..\..\..\PDF\Murata\ARRAY TYPE MULTILAYER CERAMIC Murata P1537.pdf;

– в столбце «Description Pdf» копируем из PDF строку «Multilayer Ceramic Chip Capacitor Arrays GNM Series».

7. Аналогично заполняем столбцы и для резистивных сборок и микросхемы ADS1255.

Вы вправе выбрать свою форму заполнения таблицы. Она должна полностью и адекватно отражать наполнение данного проекта, и стать в будущем основой общей базы данных для вашей организации или ваших проектов. Для других проектов вы сможете заполнять индивидуальную базу проекта не поиском, а уже ссылкой или копированием строки и общей базы компонентов.

Создание библиотеки компонентов

Итак, все компоненты будущей схемы определены и найдены их описания. Приступим к поиску и созданию компонентов и посадочных мест. Процесс создания новых компонентов описан автором ранее [1]. Здесь остановимся только на новых возможностях пакета и в применении именно данного примера.

Особенности при создании компонента:

1. Для всех компонентов введем один дополнительный параметр **L_DB** и присвоим ему значение по умолчанию «???».

2. Для всех пассивных элементов нет необходимости ввода значений параметров, характеризующих основную величину (емкость, сопротивление и т. п.) и второстепенные (напряжение, ТКЕ и т. п.). Эти параметры мы позже введем через базу данных.

3. При проектировании схемы и возможности использования нескольких посадочных мест желательно имя посадочного места на этапе проектирования отображать на схеме, что бы получить его и на твердой копии схемы. Для этого можно стандартному параметру **Comment** присвоить значение «=CurrentFootprint». Но это не всегда удобно, во-первых, потому что через данный параметр осуществляется передача других атрибутов в PCB-проект, а во-вторых, потому что, как правило, имена посадочных мест большие и занимают много места для отображения. Поэтому введем специальный параметр **Current_Footprint** (написано вмес-

те через знак подчеркивания) и присвоим ему значение «=CurrentFootprint» (написано без пробела). При этом сразу установим размер шрифта таким, чтобы текст не выходил за размеры изображения самого элемента.

4. При создании компонента и посадочного места на основе PDF из базы данных сразу внесите и название компонента в ячейку столбца «Item ID», и название посадочного места в ячейку столбца «Item Footprint» базы данных, тогда будет проще производить в будущем поиск и идентификацию их в базе.

5. При создании компонента следует учитывать также и следующие нюансы, не отраженные ранее:

– при заполнении текстовых параметров Pin можно пользоваться копированием из PDF и переносом текста через буфер обмена;

– при использовании в графике закрашенных фигур после завершения формирования компонента все закрашенные фигуры командой **Edit/Move/Send to Back** следует переместить на задний план, так как при использовании в дальнейшем преобразования схемы в формат PDF, надписи на фоне данных фигур могут не отображаться;

– настройка эквивалентности выводов и взаимозаменяемых частей компонента производится не в библиотечных редакторах, а непосредственно в схеме;

– добавлять, изменять и редактировать параметры компонентов проще через **Tools/Parameter Manager**.

Итак, создадим три компонента, необходимые для создания нашей схемы. На рис. 5 показан пример компонента резисторной сборки, для которого в панели SCH Library указано:

- **R-array_8P** — название компонента в библиотеке (принцип названия выбран следующий: краткое имя и число PIN в компоненте).
- **ARV341** — **Aliases** (другие названия этого корпуса, выбираются по названию компонента из описания) в нашем случае одно. Может быть несколько или отсутствовать.
- **R_Array_50P125X200X95-8N** — **Footprint** (название посадочного места) в нашем слу-

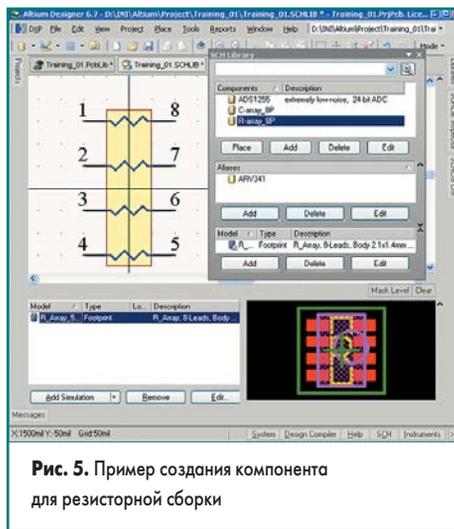


Рис. 5. Пример создания компонента для резисторной сборки

чае одно. Может быть несколько или вообще отсутствовать.

Особенности при создании посадочного места:

1. При создании посадочного места следует определиться с технологией сборки платы. В нашем примере по умолчанию мы будем создавать корпуса, используя команду **Tools/IPC Footprint Wizard** с опцией **Level B: Medium Density**. В последней версии **Altium Designer 6.7** появилась возможность автоматического создания всех корпусов проекта с помощью файла Excel с их параметрами (команда **Tools/IPC Footprints Batch Generator**). Это еще раз говорит в пользу создания и использования баз данных. Однако в данной статье мы не будем затрагивать эту проблему.

2. В посадочном месте рекомендуем сразу создавать **Component Body** для построения простых 3-мерных изображений, так как это существенно снижает вероятность ошибки наложения компонентов при размещении. Не стоит сразу заниматься поиском моделей 3-мерных изображений в библиотеках самого пакета **Altium Designer 6.7** или на сайтах производителей. Естественно, они значительно лучше отображают вид компонента, однако отнимают очень много времени и нужны только для редких проектов, требующих и механического моделирования платы.

3. В самом редакторе желательно переименовать некоторые слои, как показано на рис. 6, и назначить пары для следующих слоев:

– **Silkscreen Dimension** — схематическое изображение компонента для нанесения маркировки на плате (изображается с учетом запрета нанесения на места **PAD**). Слой **TOP Overlay** (парный слой **Bottom Overlay**).

– **Courtyard Information** — схематическое изображение зоны запрета установки других элементов для возможности монтажа и демонтажа компонента. Для этого переименовываем пару слоев типа **Mechanical в «TOP Courtyard» (Bottom Courtyard)**;

– **Assembly Information** — схематическое изображение компонента в масштабе 1:1 для оформления чертежей. Для этого переименовываем пару слоев типа **Mechanical в TOP Assy (Bottom Assy)**;

– **Component Body Information** — слой для размещения элементов для построения примитивов 3-мерного изображения. Для этого переименовываем пару слоев типа **Mechanical в TOP Component Body (Bottom Component Body)**.

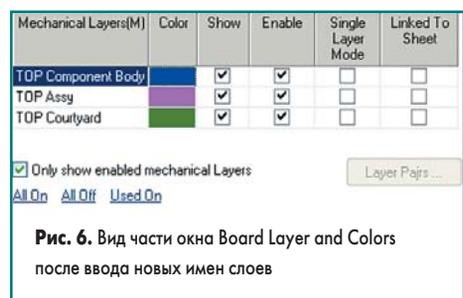


Рис. 6. Вид части окна Board Layer and Colors после ввода новых имен слоев

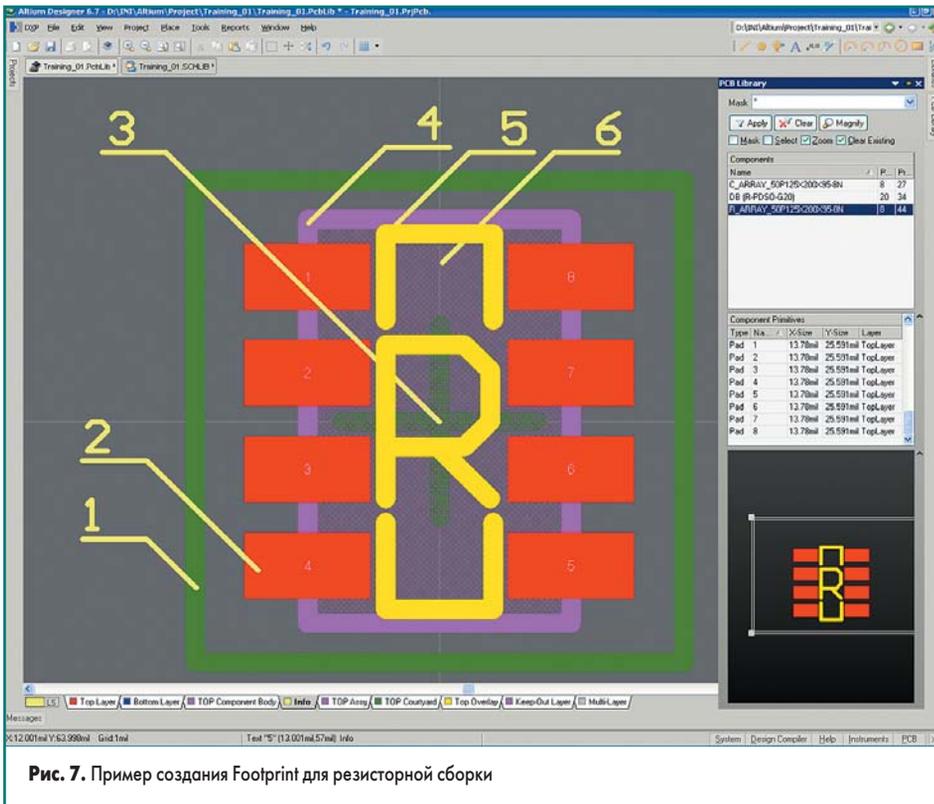


Рис. 7. Пример создания Footprint для резисторной сборки

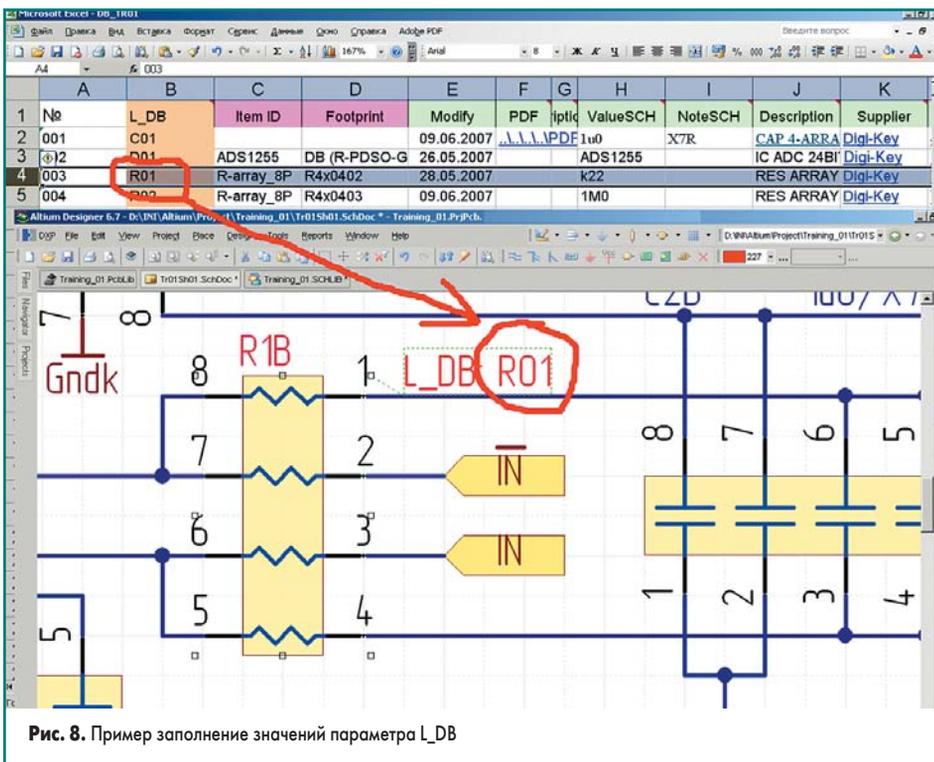


Рис. 8. Пример заполнения значений параметра L_DB

4. И, наконец, не забудьте добавить библиотеки к существующему проекту либо перетаскиванием во встроенном браузере панели **Project**, либо идентично тому, как ранее добавляли любой файл.

Итак, создадим три посадочных места, необходимые для создания проекта печатной платы. На рис. 7 показан пример посадочного места для компонента резисторной сборки **R_Array_50P125X200X95-8N**. При формировании названия посадочного места был использован следующий принцип:

- **R_Array** — название корпуса;
- **50P** — шаг между выводами;

- **125X200X95** — размеры корпуса;
- **8N** — число PAD и тип корпуса для установки на плату с плотностью **NORMAL**.

На рисунке указано:

1. **Courtyard Information** (отступы со стороны пайки больше, чем просто со стороны, где отсутствуют PAD).
2. **Pad** посадочного места.
3. Два перпендикулярных отрезка для указания центра корпуса в слое **TOP Courtyard**.
4. **Assembly Information**.
5. **Silkscreen Dimension** нанесен внутри корпуса и после его установки на плату виден не будет. Дополнительно нанесена буква

«R», для лучшей идентификации посадочного места на плате, так как точно такой же размер имеет и конденсаторная сборка.

6. Component Body Information.

Создание схемы

Схема у нас достаточно простая, и ее создание не должно вызвать особых проблем. Отразим в основном только те моменты, которые отличаются от описанных в предыдущей статье или не нашли там отражения.

1. И так все компоненты имеют один параметр: **L_DB=???**. Сделаем его видимым. После этого нам легко будет его править и контролировать.
2. При создании схемы для наших компонентов установим значение этого параметра равным значению ячейки столбца **L_DB** для строки листа базы, содержащей описание соответствующего компонента **DB_TR01.XLS**. На рис. 8 показан пример для компонента **R_Array_50P125X200X95-8N** с номиналом **220 Ом**.
3. Аналогичную операцию проделаем и для всех остальных компонентов: **R2B — L_DB=«R02»**; **C1B, C2B — L_DB=«C01»**; **D1B — L_DB=«D01»**.
4. После ввода значений параметра **L_DB** делаем этот параметр неотображаемым на схеме.
5. Командой **File/New/Library/DatabaseLibrary** создадим ссылку на файл **DB_TR01.XLS** (рис. 9). При этом выполняем последовательно операции:

А. Выбираем тип базы данных для нашего проекта **Select Database Type = Microsoft Excel**.

Б. Указываем путь, где хранится наша база проекта (файл **DB_TR01.XLS**).

В. Нажимаем кнопку **Connected**.

Г. В колонке **Table** появятся все листы из файла **DB_TR01.XLS**. Базу можно держать не только в одном листе, но и в нескольких, сгруппированных, например, отдельно для микросхем, отдельно для резисторов и т. д. В нашем случае это один общий лист **DB_TR01** для всех компонентов. И именно для него тогда и устанавливаем маркер **Enable**.

Д. В поле **Field Setting** выберем имя столбца **L_DB** для базы данных (**Database field**) и имя параметра **L_DB** для компонентов проекта (**Part parameter**). Не обязательно синхронизировать по совпадающим именам столбца и соответствующего ему параметра, можно сопоставить и разные имена столбца и параметра.

Е. С помощью команды **Tools/Option** вызовем окно свойств синхронизации с базой данных и установим свойства по умолчанию:

- **Update Values = Update** — если параметр существует для компонента проекта и есть идентичный столбец, то параметр в объекте обновить;
- **Add To Design = Do not add** — не добавлять новых параметров к проекту. (большинство параметров базы данных непосредственно в проекте не нужны);
- **Remove From Design = Do not remove** — не удалять параметры из проекта.

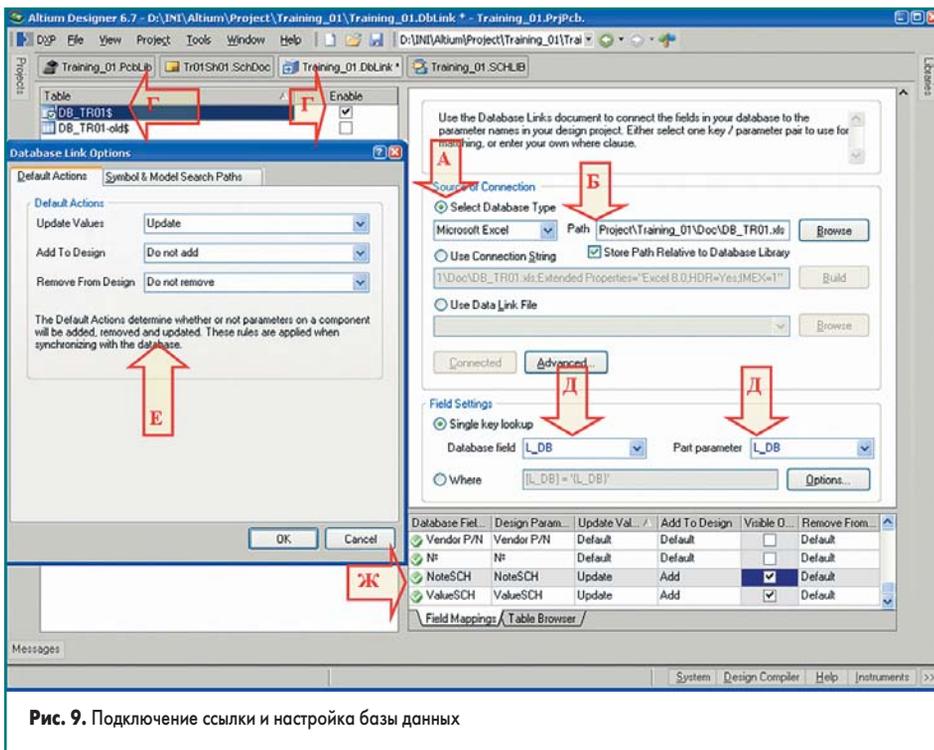


Рис. 9. Подключение ссылки и настройка базы данных

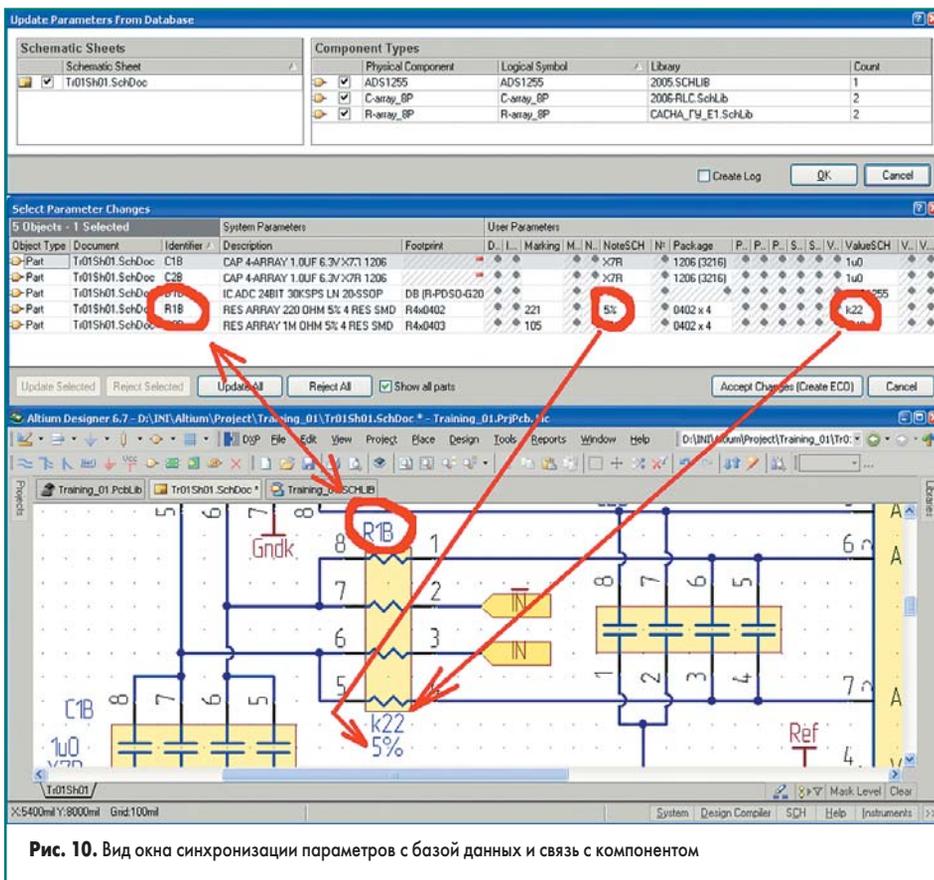


Рис. 10. Вид окна синхронизации параметров с базой данных и связь с компонентом

Ж. При выделенной строке с именем листа DB_TR01 справа внизу окна появится таблица со строками, соответствующими столбцам базы данных. На схему мы хотим добавить параметры Package (для удобства определения посадочного места), Marking (для последующей передачи данных в PCB и удобства создания вариаций монтажных схем), ValueSCH, NoteSCH и сделать видимыми ValueSCH и NoteSCH. Для них отдельно устанавливаем в соответствующей строке индивиду-

дуальные параметры синхронизации и значок отображения параметра на схеме.

- Сохраним и закроем **Traning_01.DbLink** и перейдем на окно схемы. Командой **Tools/Update Parameters From Database** вызовем окно синхронизации и введем новые параметры из базы данных. В будущем вы можете проводить синхронизацию компонентов только выбранных листов или выбранных компонентов. На рис 10 представлен вид окна и синхронизации и предложенных изменений параметров, где вы можете про-

смотреть и при необходимости исправить вносимые изменения.

- Размещаем наши отображаемые параметры на схеме так, что их было удобно читать. На рис. 10 отмечен ввод из базы данных и отображение на схеме для компонента **R1B** двух параметров **NoteSCH = «5%»** и **ValueSCH = «k22»**.

Настройка параметров проекта и его проверка

Итак, схема готова, настроим некоторые параметры проекта, которые отличаются от значений, установленных по умолчанию.

- На вкладке **Project/Project Option: Option** устанавливаем опцию **Allow Port to Name Net**, в этом случае нам не нужно присваивать имена цепям с помощью **Net Name**, что позволяет избежать дублирования идентичных имен на схеме.
- Проводим проверку листа схемы командой **Project/Compile Document Tr01Sh01.SCHDOC** (рис 11а). Проверка выдает предупреждение, что **Net Ref** не имеет источника. Вызвано сообщение тем, что вход 4 микросхемы D1B имеет тип **Input**. Ошибка не критическая, но все же открываем свойства данного компонента, входим в режим редактирования свойств **PIN** и исправляет тип данного **PIN** с **Input** на **Passive**, а отображение **PIN** делаем как **Analog Signal In** (точно также как **PIN** № 3). После этого при повторной компиляции листа схемы предупреждение, связанное с этим **PIN**, исчезнет. Желательно это исправление внести и в библиотеку.

- Проводим проверку проекта целиком командой **Project/PCB Project Tr01Sh01.PrjPCB** (рис. 11б). Проверка выдает шесть идентичных предупреждений. Ранее предупреждения не формировались, так как **Port** (а именно так произведен ввод имен цепей на листе схемы) имел свойство **Output**. При проверке проекта целиком в действительности указанные в предупреждениях **Net** не имеют источника сигнала, так как лист в этом проекте один и через **Port** к данному листу не подключена ни одна связь. Пока мы не будем обращать внимание на данные предупреждения и вернемся к этому вопросу позднее, при рассмотрении более сложных проектов, где данный лист схемы будет являться только частью общей схемы.

Создание макета печатной платы

Для нашего примера пока нет необходимости строить полный PCB проект, так как схема не имеет самостоятельного применения. Однако мы полностью проведем создание всех подготовительных действий, так как на небольшом проекте это удобно продемонстрировать. При создании первого проекта возьмем за основу модель четырехслойной платы прямоугольной формы размером 67×124,5 мм (именно этот размер будет в будущем у завершеного иерархического проекта).

- Командой **Project/Add New to Project PCB** добавляем новую PCB к нашему проекту и командой **File/Save As** сохраняем под именем **Tr01PCB01**.

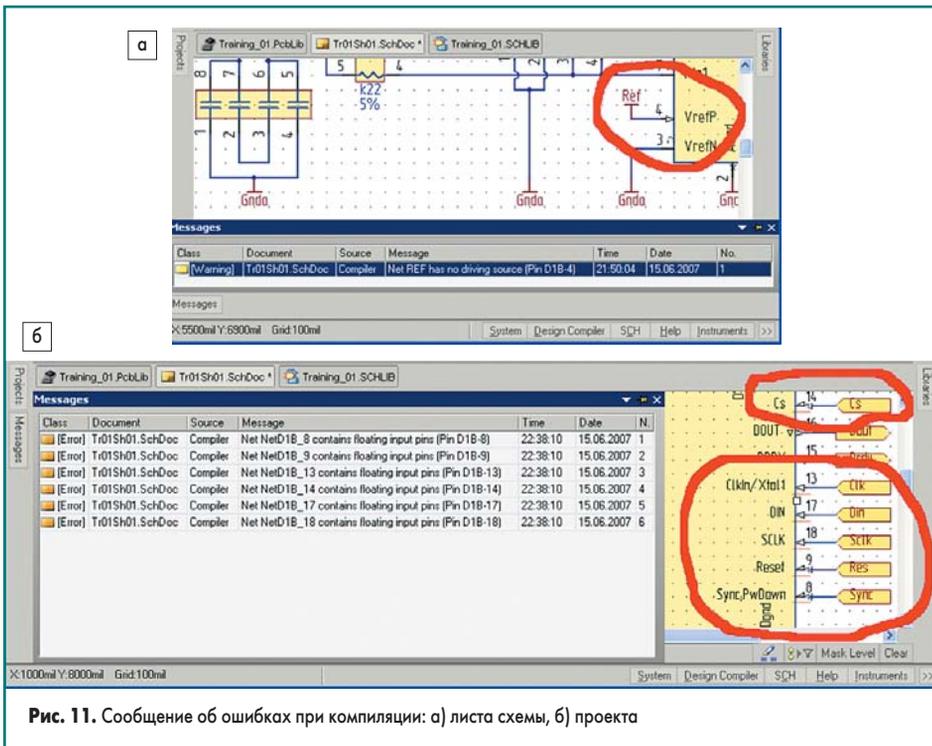


Рис. 11. Сообщение об ошибках при компиляции: а) листа схемы, б) проекта

2. С помощью команды **Design/Board Option** вызываем окно параметров PCB-проекта и там устанавливаем:

- **Measurement Units = Metric** — для ведения всех измерений в миллиметрах.
- **Sheet Position** — максимальные размеры отображаемого поля. В нашем примере всю документацию будем оформлять на листе формата A4, поэтому указываем его размеры в соответствии с ГОСТ.
- Параметры всех сеток устанавливаем под свои требования.

3. Командой **Design/Layer Stack Manager** вызываем окно настройки параметров физических слоев будущей платы. К этому моменту мы должны определиться со структурой слоев, а также технологий при производстве платы — толщиной фольги для каждого слоя и типом и толщиной межслойного заполнителя. Типовые варианты значений вы должны выяснить у производителя печатных плат. В данном примере будут отсутствовать слои типа **Plane**.

А. Для добавления двух внутренних слоев нажимаем кнопку **Add Layer** и вводим название нового слоя.

Б. Для всех слоев и межслойных заполнителей указываем их свойства (толщина для фольги, толщина и диэлектрическая проницаемость для заполнителя и защитного покрытия). Выбор толщин производим так, чтобы общая толщина печатной платы для нашего проекта была равной примерно 1,5 мм.

4. Точно так же, как при создании посадочных мест, переименуем механические слои и образуем из них пары (рис. 12):

- **TOP Courtyard** парный с **Bottom Courtyard**;
- **TOP Assy** парный с **Bottom Assy**;
- **TOP Component Body** парный с **Bottom Component Body**.

5. Переименуем еще три механических слоя в (рис. 12):

- **Board Outline** — слой внешнего контура печатной платы;

- **Note** — слой, где поместим специализированные требования для производства плат и другую техническую информацию;
- **Template** — слой для размещения рамки и общих текстовых надписей, необходимых для чертежей.

6. Отображение, цвет и другие настройки всех слоев настраиваем индивидуально (рис. 13).

7. На слое **Template** наносим рамку шаблона листа A4 и заполняем все поля рамки, которые будут иметь одинаковое значение для всех технических чертежей.

8. На слое **Board Outline** наносим прямоугольный контур платы.

9. Выделяем контур **Board Outline** и командой **Design/Board Shape/Redefine Board Shape from selected objects** выделяем цветом область размещения платы.

10. На слое **Keep-Out Layer** наносим прямоугольный контур внутри которого должны быть размещены как посадочные места, так и элементы трассировки. В нашем случае он будет совпадать с контуром на слое **Board Outline** (мы не будем ограничивать область для размещения компонентов, дорожек и т. п. на плате).

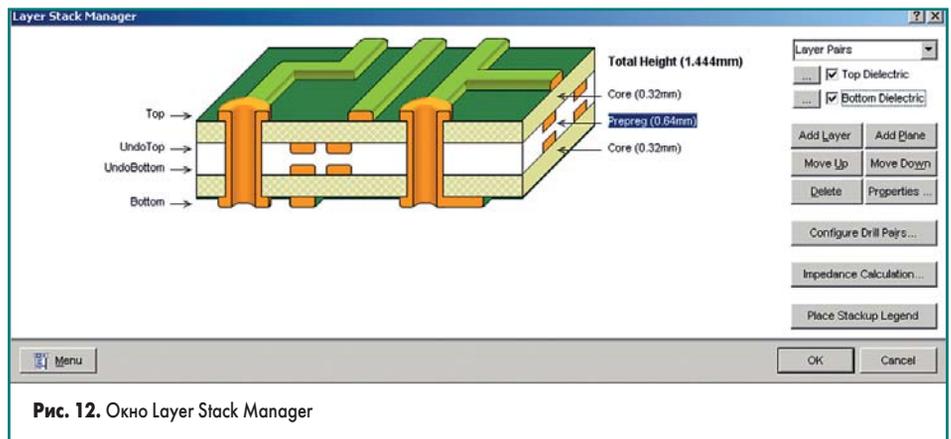


Рис. 12. Окно Layer Stack Manager

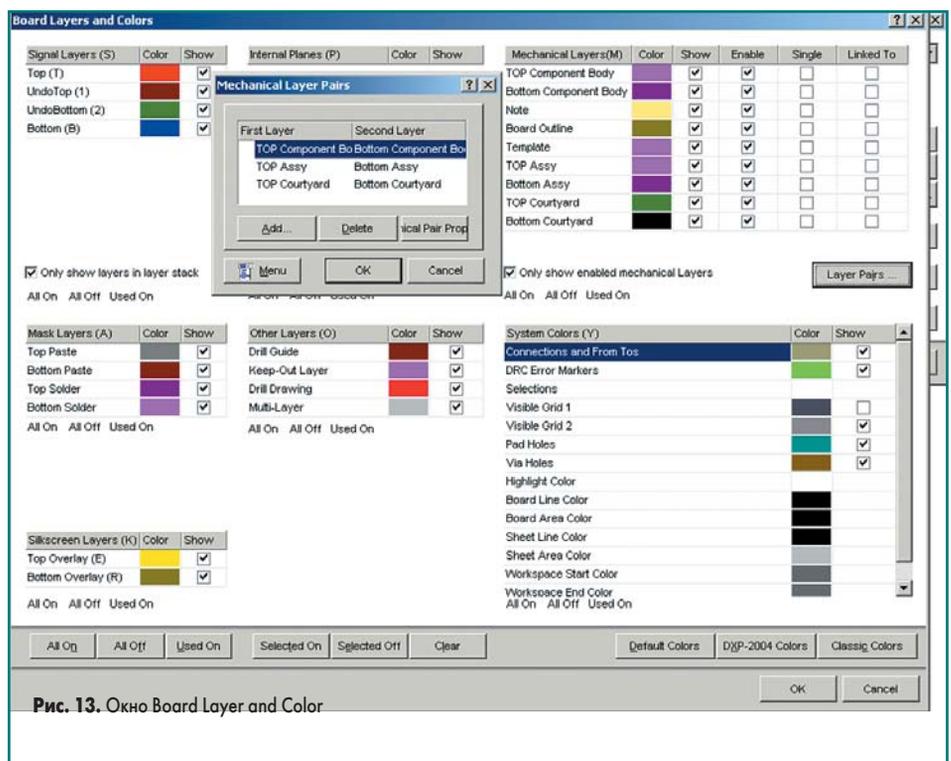


Рис. 13. Окно Board Layer and Color

11. На слое Note:

- используя команду **Place/Dimension/Datum** наносим размеры контура печатной платы;
- в рамках листа A4 на свободном месте командой **Tools/Layer Stackup Legend** размещаем таблицу со сведениями о технологических слоях.
- Под таблицей дополнительно наносим сведения:
 - a. **Gerber Note Layer (.GNL)**, «**Layer_Name**» (отобразится надпись «**Note**») — сведения о текущем слое и расширении в имени файла GERBER текущего слоя;
 - b. **Gerber Board Outline (.GBO)**, «**Board Outline**» — сведения о расширении в имени файла GERBER для контура платы;
 - c. **NC Drill (.TXT)** — сведения о расширении в имени файла для станка при сверловке отверстий (без разделения на металлизированные и не металлизированные отверстия);
 - d. **NC Drill Plated (_-plated.TXT)**, **NC Drill NoPlated (_-noplated.TXT)** — сведения об окончании в имени файла для станка при сверловке отверстий (для металлизированных и не металлизированных отверстий, для некруглых отверстий);
 - e. аналогично, если в проекте присутствуют фрезерованные и другие специализированные отверстия.

12. На слое Drill Drawing располагаем текст **Place/String = «Legend»** (вначале слова обязательно точка) непосредственно над платой. При формировании **Drill Drawing** специальными средствами на этом месте будет располагаться таблица сведений о параметрах и количестве отверстий в печатной плате. В зависимости от количества строк, таблица будет иметь неопределенную высоту, поэтому сверху от этой надписи должно быть зарезервировано место, и других надписей и обозначений в этой зоне располагать не рекомендуется.

13. Справа от платы расположим сверху вниз на соответствующем слое их название и расширение названия в файле GERBER, в этом случае названия будут нанесены прямо на GERBER и производителю будет легче ориентироваться в назначении конкретного фотосаблона:

- **.GTP «Layer_Name»** (отобразится как **Top Paste**) — GERBER для изготовления шаблона нанесения пасты для пайки;
- **.GTO «Layer_Name»** (отобразится как **Top Overlay**) — GERBER для изготовления шаблона нанесения надписей;
- **.GTS «Layer_Name»** (отобразится как **Top Solder**) — GERBER для нанесения защитной маски;
- **.GTL «Layer_Name»** (отобразится как **Top**) — GERBER верхнего слоя печатной платы;
- **.G1 «Layer_Name»** (отобразится как **UndoTop**) — GERBER первого внутреннего слоя печатной платы;
- **.G2 «Layer_Name»** (отобразится как **UndoBottom**) — GERBER второго внутреннего слоя печатной платы;

- **.GBL «Layer_Name»** (отобразится как **Bottom**) — GERBER нижнего слоя печатной платы;
- **.GBS «Layer_Name»** (отобразится как **Bottom Solder**) — GERBER для нанесения защитной маски;
- **.GBO «Layer_Name»** (отобразится как **Bottom Overlay**) — GERBER для изготовления шаблона нанесения надписей;
- **.GBP «Layer_Name»** (отобразится как **Bottom Paste**) — GERBER для изготовления шаблона нанесения пасты для пайки;

14. Стандарт IPC-7351(L) требует установки на печатной плате трех позиционных меток **Global Fiducials** с радиусом от 1 до 3 мм со стороны установки элементов (рис. 14). Данные метки могут использоваться в технологических целях. В дальнейшем мы будем ориентироваться на двухсторонний монтаж, поэтому нам нужно установить 3 метки на слое **TOP** и 3 метки на слое **Bottom**, и сделать зону запрета трассировки и размещения элементов возле этих меток. Данные метки должны быть установлены после трассировки в свободных местах. Поэтому мы установим за пределами платы в левом нижнем углу заготовку такой метки, которую потом скопируем и разместим после трассировки платы в свободных местах.

- Командой **Place/Pad** ставим в это место **PAD** и устанавливаем ему свойства: **Designator=GF** (аббревиатура от **Global Fiducials**); **Layer=TOP**; **Net=NoNet**; **X-Size=Y-Size=1mm**; **Shape=Round**.
- Командой **Place/FullCircle** рисуем вокруг данного **PAD** на слое **TOP** круг с радиусом 2mm и устанавливаем ему свойства: **Net=NoNet**; **KeepOut=True**. Последний параметр означает, что данный круг будет отображаться на соответствующем слое, но не будет передаваться в GERBER.
- Делаем копию данного **PAD** и круга, переводим копии в слой **Bottom** и совмещаем с **PAD** и кругом на слое **TOP**.
- Выделяем совмещенные два **PAD** и два круга, вызываем щелчком правой кнопки мышки контекстное меню. Командой **Unions/Create Union from selected objects**

создаем **Union**. Теперь наш комплексный элемент из двух **PAD** и двух окружностей будет перемешаться как единое целое.

15. Наконец, поставим на печатную плату крепежные отверстия. Печатная плата должна крепиться к металлическому корпусу с помощью винтов с резьбой M3 и через них должна и подключаться корпусная «земля». Крепежные отверстия должны быть не металлизированы, так как при вращении винта металлизация может быть оторвана от платы и **PAD** такого отверстия. Соединение металлизированного ободка на слоях произведем несколькими **VIA**, расположенными равномерно на ободке.

- Командой **Place/Pad** располагаем **PAD** и устанавливаем ему свойства:
 - a. **Designator=M6-3.3**;
 - b. **Layer=Multi-Layer** (сквозное отверстие);
 - c. **X-Size=Y-Size=6mm** (больше размера шайбы под винт);
 - d. **Hole Size= 3.3mm** (рекомендуемый размер для винта с резьбой M3);
 - e. **Shape=Round**;
 - f. **Net=GndK** (**Pad** должен быть подключен к корпусной «земле»);
 - g. **Plated=False** (не металлизированное отверстие).
- Командой **Place/Via** располагаем четыре (или больше) переходных отверстия на ободке **Pad**.
- Выделяем совмещенные **PAD** и все **Via** и создаем из них **Union**, как было описано в предыдущем пункте.
- Устанавливаем **Union** в угол платы, копируем его и добавляем в оставшиеся углы платы.
- Используя команду **Place/Dimension/Datum**, наносим размеры контура печатной платы, включая и крепежные отверстия. После этого перемещаем наши **Union** на нужные координаты, **Datum Dimension** при этом будет отслеживать перемещения, постоянно отображая текущие расстояния.

Подготовка проекта платы завершена. Выделяем все элементы и устанавливаем опцию **Lock** для защиты от случайных переме-

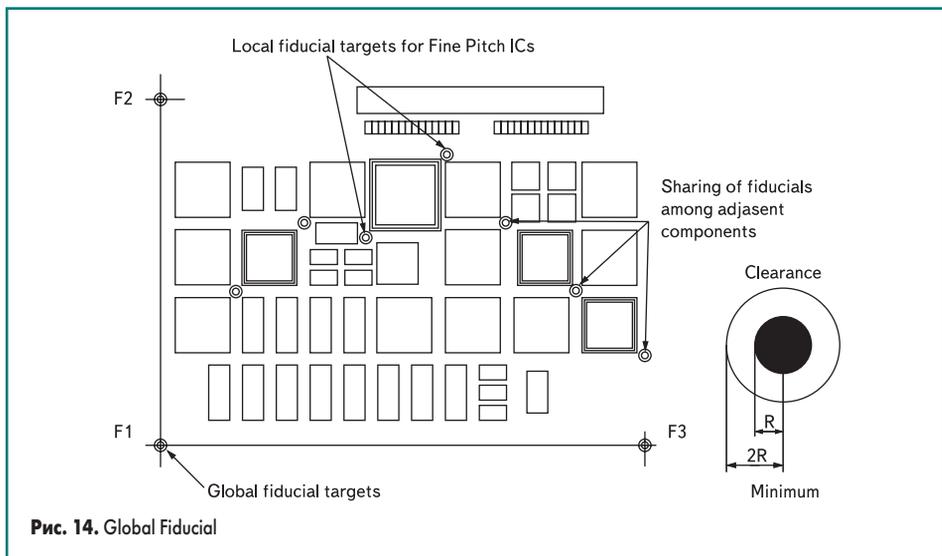


Рис. 14. Global Fiducial

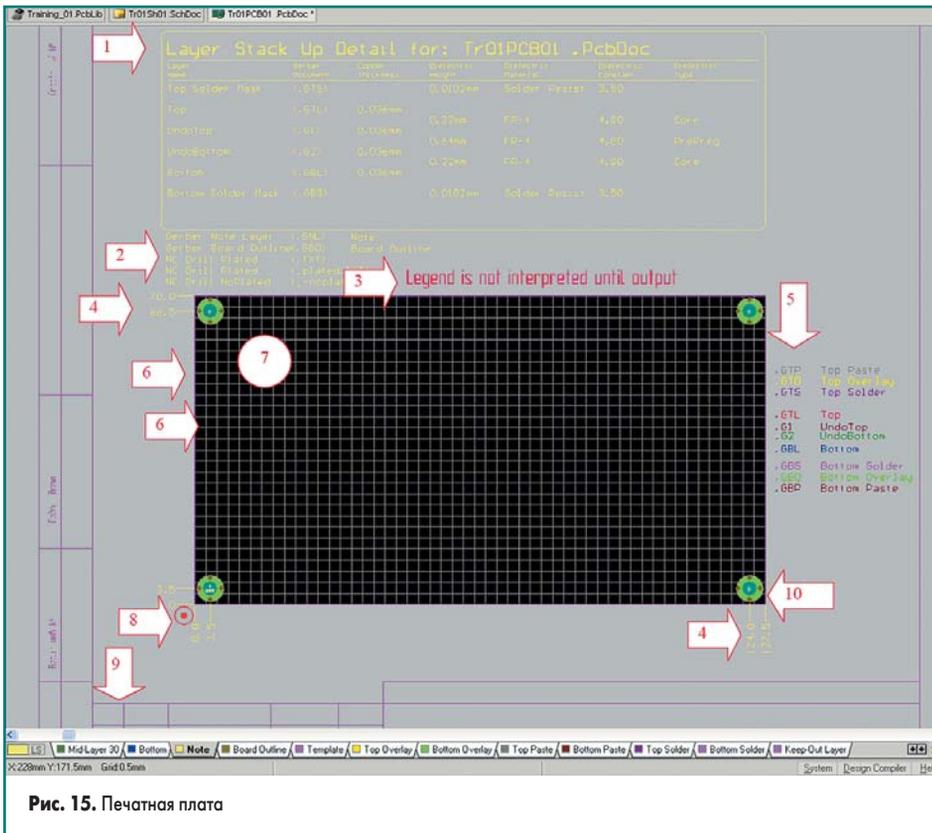


Рис. 15. Печатная плата

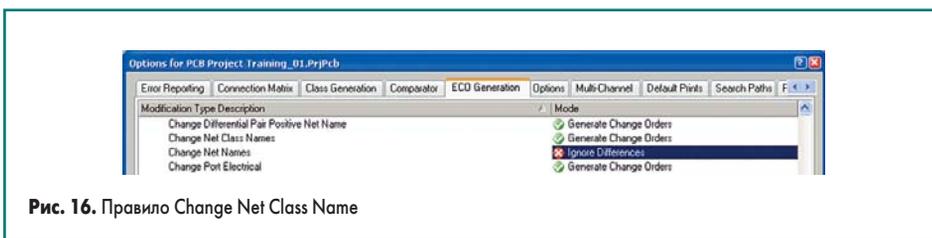


Рис. 16. Правило Change Net Class Name

шений. Результат представлен на рис. 15, где отмечено:

1. зона размещения **Layer Stackup Legend**;
2. зона размещения информации о дополнительных **Gerber** и **NCDrill**;
3. зона размещения информации ссылки на таблицу **Drill Drawing** (на рис. 15 сознательно опущена свободная зона между **Layer Stackup Legend** и **Drill Drawing**);
4. **Datum Dimension** — указание размеров контура платы;
5. зона размещения информации о расширении названий файлов **Gerber** и их названий;
6. **KeepOut** — контур печатной платы и контур;
7. зона печатной платы;
8. заготовка одного **Global Fiducial**;
9. рамка по ГОСТ для формата A4;
10. крепежное отверстие.

Перед чем начать трассировку платы, создадим типовые классы и правила.

Создание классов для печатной платы

Перед созданием классов следует передать все объекты из схемы на PCB. Для этого из PCB проекта используем команду **Design/Import Changes from Source Documents**. Чтобы при будущей синхронизации схемы и PCB-проекта, вновь добавленные классы не удалялись из проекта командой **Design/Option: ECO Generator** изменим правило для ECO-генератора как указано на рис 16.

Командой **Design/Classes...** открываем окно **Object Class Explorer** и далее:

1. **Net Classes**. Дополнительно к стандартному классу **All Nets** создадим еще два:
 - **Power**, куда включим **Net (+3.3V, NetD1B)**;
 - **GND**, куда включим **Net (Gnd, GndA, GndK)**.
2. **Component Classes, Layer Classes**. Новых классов создавать не будем.
3. **PAD Classes**:
 - **PAD_GF** для **PAD**, принадлежащих **Global Fiducial (Free-GF)**;
 - **PAD_Mount** для **PAD**, принадлежащих крепежным отверстиям (**Free-M6-3.3**).
4. Для оставшихся в списке типов новых классов создавать не будем.

Создание правил для печатной платы

Перед созданием правил определимся с классом печатной платы по ГОСТ 23751-86. На данный момент самым меньшим посадочным местом являются резисторные и конденсаторные сборки с шагом выводов 0,5 мм и шириной PAD 0,35 мм. Таким образом, расстояние между ними (зазор) равно 0,15 мм, что соответствует 4 классу. Поэтому на данный момент будем задавать все правила исходя из 4 класса печатной платы:

- минимальные зазоры между проводящими элементами — 0,15 мм;
- минимальная ширина дорожек — 0,15 мм;

- гарантированный поясок для отверстий — 0,05 мм;
- значение наименьшего диаметра металлизированного отверстия определим из расчета $1,5/4=3,75$ мм и с учетом толщины металлизации, допустимых отклонений и номинального ряда диаметров сверла устанавливаем равным 0,5 мм.

Командой **Design/Rule** открываем окно **PCB Rule and Constraints Editor** и далее создаем правила, приведенные ниже (приоритет первого в списке для каждого правила самый высокий, последнего — самый низкий). Для задания или изменения приоритета правил внизу окна **PCB Rule and Constraints Editor** имеется кнопка **Priorities...**

1. **Electrical** (правила приведены в порядке приоритетов от высшего к низшему):

- **Clearance** (правило для зазоров).
 - a. Вводим новое правило с именем **Clearance_PAD_to_PAD** для задания зазора (0,15 мм) между **PAD**;
 - **Where the First object matches = (IsPAD)**;
 - **Where the Second object matches = (IsPAD)**;
 - **Minimum Clearance = 0.149 мм**.
 - b. Вводим новое правило с именем **Clearance_Power_to_GND** для задания увеличенного зазора (0,25 мм) между сигналами класса **Power** и класса **GND**, где указываем:
 - **Where the First object matches = InNetClass('Power');**
 - **Where the Second object matches = InNetClass('GND');**
 - **Minimum Clearance = 0.25 мм**.
 - c. Вводим новое правило с именем **Clearance_Power_or_GND_to_All** для задания зазора (0,2 мм) между сигналами классов **Power** и **GND** и остальными цепями, где указываем:
 - **Where the First object matches = (InNetClass('Power') OR InNetClass('Gnd'));**
 - **Where the Second object matches = All;**
 - **Minimum Clearance = 0.2 мм**.
 - d. Для стандартного правила **Clearance** устанавливаем **Minimum Clearance = 0.15 мм**.

2. **Routing** (правила приведены в порядке приоритетов от высшего к низшему):

- **Width** (правило ширины дорожек).
 - a. Вводим новое правило с именем **GNDK**:
 - **Where the First object matches = InNet('GNDK')** — только для указанной цепи;
 - **Min Width = 0.3 мм**, — минимальная ширина;
 - **Preferred Width = 1 мм**, — ширина по умолчанию;
 - **Max Width = 3 мм**, — максимальная ширина.
 - b. Вводим новое правило с именем **GND**:
 - **Where the First object matches = InNet('GNDK');**
 - **Min Width = 0.25 мм**,
 - **Preferred Width = 1 мм**,
 - **Max Width = 1.5 мм**;

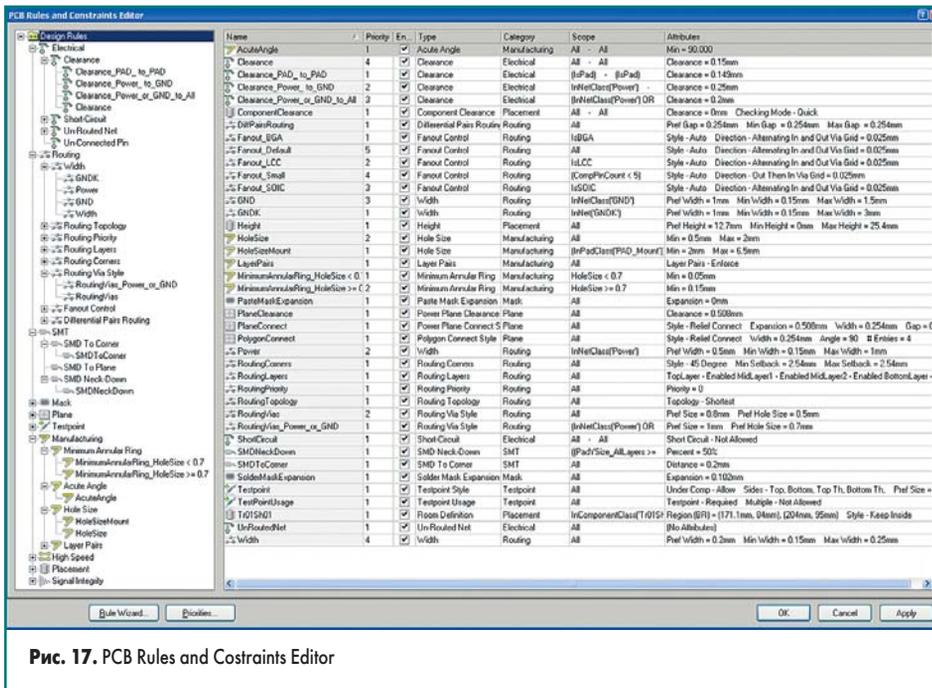


Рис. 17. PCB Rules and Costraints Editor

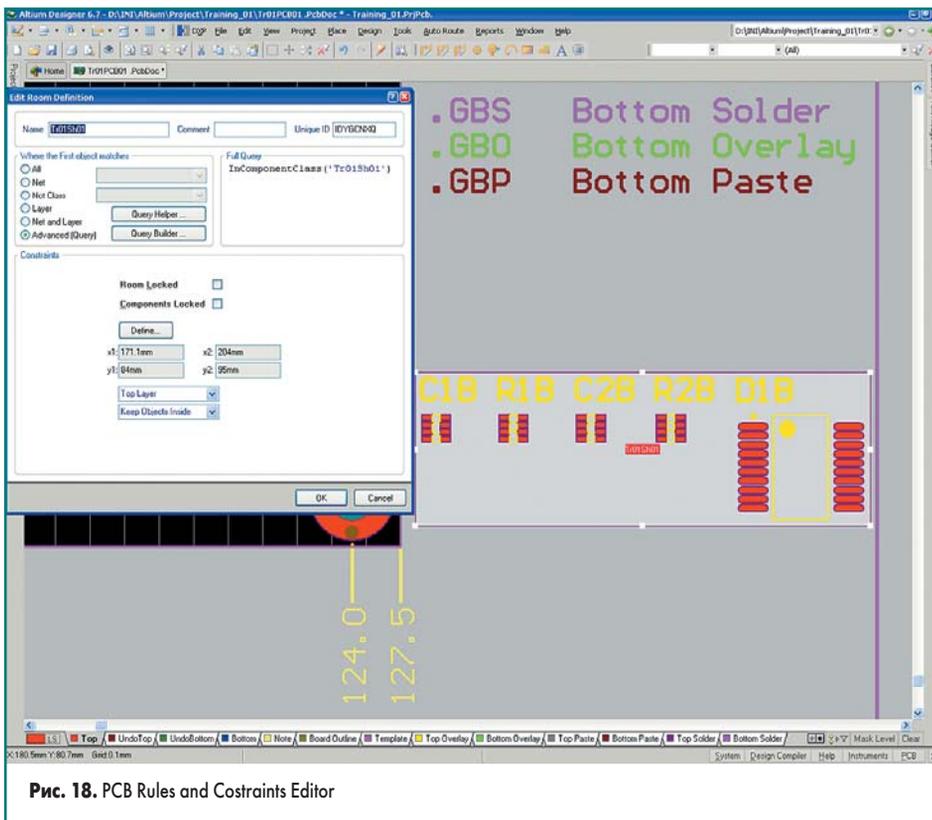


Рис. 18. PCB Rules and Costraints Editor

Distance= 0.2 мм — так, что бы первый излом начинался уже по маской.

– SMDNeckDown (подвод к SMD PAD с опцией сужения дорожки).

- Where the First object matches = ((PadYSize_AllLayers >= PadXSize_AllLayers) And (PadXSize_AllLayers > 0.6)) Or ((PadXSize_AllLayers >= PadYSize_AllLayers) And (PadYSize_AllLayers > 0.6));
- Neck=Down=51% (сужение для всех PAD, один из размеров которого больше 0.6 мм).

4. Manufacturing (технологические правила):

- MinimumAnnularRing (гарантированная ширина ободка):
 - a. MinimumAnnularRing_HoleSize < 0.7, Minimum Annular Ring = 0.05 mm (для переходных отверстий);
 - b. MinimumAnnularRing_HoleSize >= 0.7, Minimum Annular Ring = 0.15 mm (для относительно больших отверстий).

– Acute Angle (значение острого угла при между дорожками)

- a. AcuteAngle, Minimum Angle = 90 (запрет острых углов).

– Hole Size (диапазон значений диаметра отверстия)

- a. HoleSizeMount (для монтажных отверстий);
 - Where the First object matches = (InPadClass('PAD_Mount'));
 - Measurement method = absolute;
 - Minimum = 2 mm; Maximum = 6.5 mm.
- b. HoleSize (для всех отверстий);
 - Where the First object matches = All;
 - Measurement method = absolute;
 - Minimum = 0.5 mm; Maximum = 2 mm.

5. Placement (правила для размещения компонентов):

- ComponentClearance (расстояние между компонентами).
 - Where the First object matches = All;
 - Where the Second object matches = All;
 - Check Mode = Quick Check — (формально по наибольшему размеру, что соответствует контуру на слоях Courtyard;
 - Minimum Horizontal Gap=0 mm.

На этих правилах остановимся, их список представлен на рис. 17.

Размещение и разводка

При импорте из схем командой Design/Import Changes from Source Documents все элементы располагаются справа (рис. 18) от печатной платы и формируется Room (в нашем случае Tr01Sh011 или по имени листа схемы). На рисунке Room показано выделение Room и его свойства.

«Перетаскиваем» их на поле печатной платы, при этом компоненты переместятся вместе с Room. Выделив Room, зададим ей нужный размер, передвигая белые квадратики в углах Room и в середине сторон (рис. 19). Если нужно задать более сложную форму Room используйте команду Design/Rooms/Move Polygonal Room Vertices. При этом белые квадратики в углах можно передвинуть в любое относи-

- c. Вводим новое правило с именем Power:
 - Where the First object matches = InNetClass('Power');
 - Min Width = 0.25 mm, Preferred Width = 0.5 mm, Max Width = 1 mm;
 - d. Для стандартного правила с именем Width:
 - Where the First object matches = All;
 - Min Width = 0.15 mm, Preferred Width = 0.2 mm, Max Width = 0.25 mm.
- Routing Via Style (правила для переходных отверстий).
- a. Вводим новое правило с именем RoutingVias_Power_or_GND для раз-

- водки земли и шин питания с увеличенными переходными отверстиями:
- Where the First object matches = (InNetClass('Power') OR InNetClass('Gnd'));
 - Via/Hole Diameter Min = 0.7/0.5 mm, Preferred = 1/0.7 mm, Max = 1.5/1 mm;
- b. Для стандартного правила с именем RoutingVias;
 - Where the First object matches = All;
 - Via/Hole Diameter: Min= 0.7/0.5 mm; Preferred= 0.8/0.5 mm, Max= 0.9/0.5 mm.
3. SMT (правила для компонентов поверхностного монтажа):
- SMDToCorner (расстояние до первого излома). Where the First object matches = All,

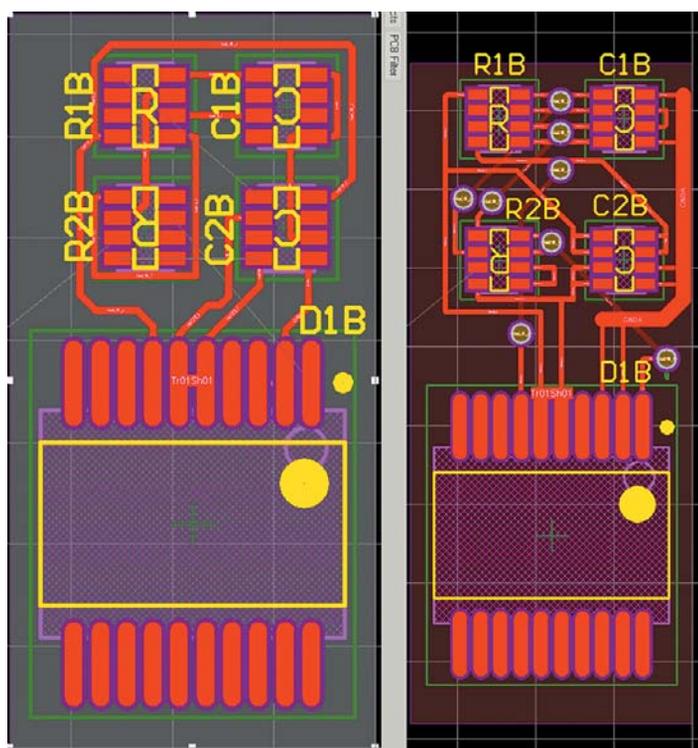


Рис. 19. Результат автоматической и интерактивной трассировки Room

тельное место, а белые квадратики на сторонах **Room** добавляют новый угол (**Vertice**) **Room**.

Командой **Design/Component Placement/Arrange Components Within Room** поместим все компоненты внутри **Room**. При написа-

нии правил мы практически полностью пропустили раздел **Placement**, поэтому запуск авторасстановщика командой **Design/Component Placement/Auto Placer** не даст хороших результатов, но пять компонентов легче, а главное, проще и быстрее расположить вручную,

чем писать сложные правила расстановки. Расположим компоненты вручную внутри **Room** и при необходимости корректируем его форму.

При выделенной **Room** через контекстное меню командой **Room Action/Autorotation Room** производим автотрассирование **Room**. Результат представлен на рис. 19 слева и нас не устраивает. Действительно мы всерьез не занимались правилами для автотрассирования и поэтому получили такой результат. Нас он не удовлетворяет и рядом на рис. 19 справа представлен результат интерактивной трассировки печатной платы. Как видно из рис. 19, пришлось увеличить расстояние между посадочными местами и использовать дополнительный слой.

В данном примере осталось около 10 связей, которые подключены только к одному **PAD** и одна не разведенная связь **GNDK**, которую, впрочем, мы и не пытались развести, так как она находится и за пределами **Room** (крепежные отверстия).

На этом рассмотрение первого примера завершим. В следующем примере рассмотрим пример реализации 8- и 16-канальных АЦП на основе данной схемы.

Продолжение следует

Литература

1. Пранович В. Система проектирования Altium Designer 6 // Технологии в электронной промышленности. 2006. № 5, 6. 2007. № 2.