Продолжение. Начало в № 5`2007

Altium Designer 6 в примерах

Владимир Пранович, к. т. н.

pranovich@bsu.by

Пример 3. Модуль 16-канального АЦП со схемой управления и передачи данных

На основе проекта, использованного в данном примере, рассмотрим особенности создания сложных библиотечных элементов, а также подходов при разработке схемы и создании топологии. Эти подходы способны вызвать определенный интерес и могут быть использованы и в других проектах. Вся схема целиком в статье приведена не будет, мы представим лишь отдельные ее фрагменты, на которых остановимся подробнее.

Аналогично, как было сделано в предыдущем примере, создадим для следующего проекта отдельную папку и переименуем все файлы, как делали ранее. Только теперь учтем прежний опыт и:

1. Зарезервируем название Tr04Sh01.SchDoc для первого листа, который оставим пока незаполненным. На данном листе в будущем разместим только типовые элементы, встречающиеся во всех схемах — это стабилизаторы питания, фильтровые конденсаторы и ссылки на подчиненные листы схемы.
2. Название Tr04Sh02.SchDoc присвоим листу общей части схемы, относящейся к 16-канальному АЦП.
3. Переименуем и соответственно изменим на схеме ссылки на подчиненные листы.
   – Tr04Sh01.SchDoc => Tr04Sh03.SchDoc. Это лист с общей частью 8-канального АЦП. Изменим для этого листа файл шаблона, а на самой схеме — ссылку на подчиненный лист с Tr03Sh02.SchDoc => Tr04Sh04.SchDoc. Остальную часть схемы оставим без изменений.
   – Tr04Sh02.SchDoc => Tr04Sh04.SchDoc. Это схема 8-канального АЦП. Она также взята без изменений.
   – Остальные листы, начиная с пятого, зарезервируем под схемы микроконтроллера, передачи данных и канала управления.
4. На второй лист, как делали в предыдущем примере, поместим ссылку (Sheet Symbol) на подчиненный (третий) лист схемы Tr03Sh03.SchDoc.
5. На второй лист добавим схемную часть, касающуюся общей части для 16-канального листа.
6. На пятый лист поместим ссылку на подчиненные (второй, шестой и седьмой) листы схемы Tr03Sh02.SchDoc, а на первый лист — ссылку на подчиненный (пятый) лист схемы.
7. Аналогично, как в предыдущем примере, произведем перенумерацию листов и другие действия.
8. Скомпилируем проект.

Результат действий представлен на рис. 35. Необходимо обратить внимание на следующие особенности:

1. Структура основных файлов проекта. Иерархия проекта характеризуется соответствующим видом дерева файлов.
2. Вспомогательные файлы проекта (подключенные библиотеки и базы данных).
3. Информационные файлы (описания и т. п.).
4. Часть первого листа с **Sheet Entry** (ссылка) на лист схемы микроконтроллера.
5. Лист схемы микроконтроллера.
6. **Sheet Entry** на листе схеме микроконтроллера на листы схемы 16-канального АЦП, интерфейса RS422 и трансивера E1.
7. Чистые листы для схем интерфейса RS422 и трансивера E1.
8. Лист со схемой 16-канального АЦП.
9. Два **Sheet Entry** на лист схемы с 8-канальным АЦП.
10. Схема 8-канального АЦП.
11. Многоканальный **Sheet Entry** на лист схемы с каналом АЦП.
12. Схема с каналом АЦП.
13. Указано 16 окон схемы канала АЦП, в частности открыт лист, где приведено обозначение канала A14.

Теперь рассмотрим подробней представленный результат и покажем, каким образом он достигнут.

Лист со схемой канала АЦП (выноска 12 на рис. 35) не претерпел никаких изменений по сравнению со схемой предыдущего примера.

На листе **Tr04Sh03.SchDoc** с общей частью 8-канального АЦП (рис. 35, выноска 10) внесены изменения в схему дешифратора с целью получения кодирования для 16 каналов. Они отмечены выноской **A** на рис. 36.

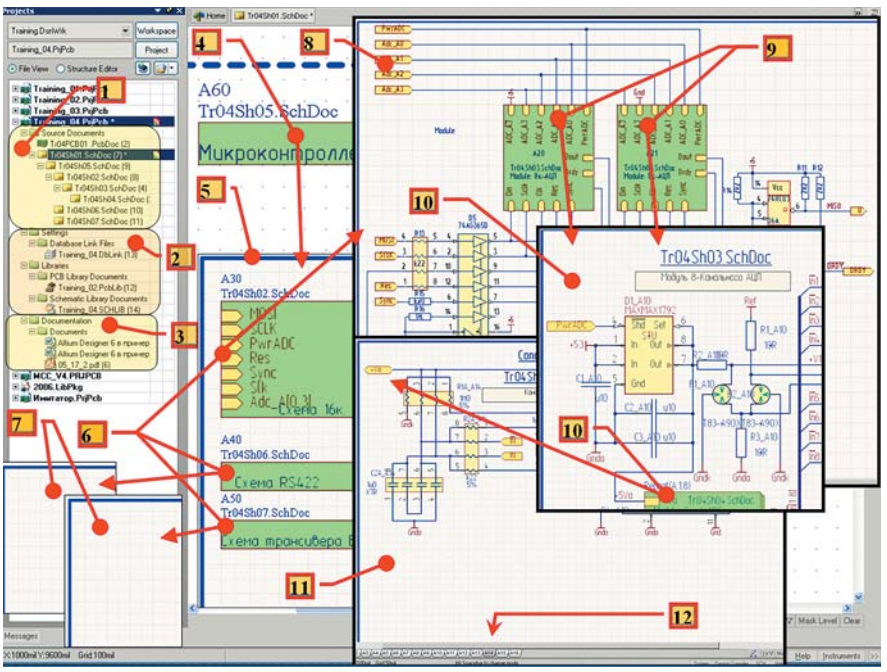

Рис. 35. Вид проекта и связи между листами

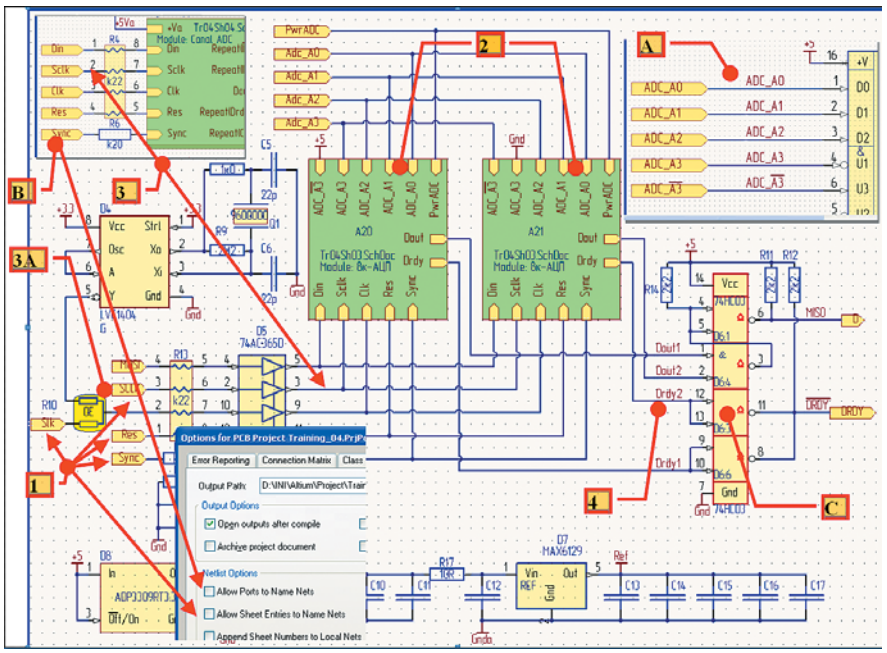

Рис. 36. Схема 16-канального модуля АЦП

Рассмотрим вопросы связанности листов схем в данном проекте (рис. 36):

1. На выноске **B** представлена часть подчиненного листа **Tr04Sh03.SchDoc** с приведенными **Port** с названиями **Din**, **Sclk**, **Clk**, **Res**, **Sync**, полностью совпадающими с аналогичными на листе **Tr04Sh02.SchDoc**. Однако физически это разные электрические связи. Поэтому в данном проекте следует запретить присваивание электрическим цепям имен по названиям **Port** и **Sheet Entry**.
2. При создании схемы данного проекта (16-канальный АЦП) необходимо дважды повторить схему 8-канального АЦП. Такое решение использовано и для демонстрации другой возможности: в схеме использованы два **Sheet Symbol** со ссылкой на один и тот же лист 8-канального АЦП.
3. Таким образом, сигнал **Sclk**, отмеченный соответствующим **PORT** на листе **Tr04Sh03.SchDoc**, не имеет **NetLabel** на листе **Tr04Sh02.SchDoc**, и, конечно же, это будет совершенно другой физический сигнал, нежели сигнал, отмеченный как **Port** = **Sclk** (выноска 3A).
4. Адресация с 1-го по 8-й и с 9-го по 16-й каналы обеспечена подачей соответствующих сигналов на **Port** для прямого и инверсного входов дешифраторов, совпадающие входные цепи которых объединены, а выходы связаны, как указано на схеме.

Остановимся подробней на компоненте 74HC03 (четыре идентичных двухвходовых логических элемента). В данном примере рассмотрим его применение с точки зрения реализации функции взаимной замены, как идентичных входов логических элементов, так и самих логических элементов, размещенных в данном корпусе.

Такой компонент в нашем случае будет состоять, например, из 6 **Part**, четыре из которых одинаковы (логические элементы), а два содержат только по одному **Pin** (для подключения «земли» и питания соответственно). Такое решение обеспечивает простоту присоединения последних двух **Part** к одному или группе идентичных, при этом там, где это удобно на схеме. Более того, такое решение гарантирует, что при перенумерации не будет происходить переноса из изображений **Pin** для питания, если бы мы их присвоили одному из логических элементов. В принципе **Pin** для питания вообще можно не отображать на схеме, но автор — не сторонник такого подхода. Итак, при создании данного компонента в библиотеке необходимо обратить внимание на следующие особенности (рис. 37):

1. Добавление нового **Part** к компоненту производится из выпадающего меню **Place**.
2. Взаимозаменяемые **Part** должны иметь одинаковое положение относительно центра листа в библиотеке.
3. Для незаменяемых **Part** в нашем случае для подключения питания и «земли» следует указать свойство **Locked**. При автонумерации на схеме не будет изменяться номер **Part** данного компонента.
4. Командой **Tools/Configure Pin Swapping** вызываем окно **Configure Swapping Information In Component**, выделяем компонент (74HC03_4G+Vcc+Gnd), в котором разрешим взаимную замену **Part** (логических элементов) и **Pin** (идентичных входов этих логических элементов). После этого нажимаем кнопку **Configure Component**. **Pin** с одинаковыми кодами (номерами) в соответствующей колонке могут быть взаимно заменены при помощи команды **Swap…**. То же касается и **Part**.
   Примечание. *Если вы не находите необходимый компонент, снимите отметку у* ***Only Show Component with Swapping Information****, и в таблице появятся строки со всеми компонентами библиотеки.*
5. В окне **Configure Pin Swapping for…** в колонке **Pin Group** указываем эквивалентность **Pin**. В нашем примере в **Part** (логическом элементе) только одна группа эквивалентных выводов — это входы логического компонента. Поэтому для них в соответствующей колонке устанавливаем значение «единица». При использовании более сложных компонентов с несколькими группами взаимозаменяемых **Pin** следует каждой группе идентичных **Pin** присвоить свой уникальный код.
6. В колонке **Part Group** аналогично указываем для взаимозаменяемых **Part** одинаковый код (в нашем случае «единица»), для незаменяемых (два **Part**, содержащих **Pin=Vcc**

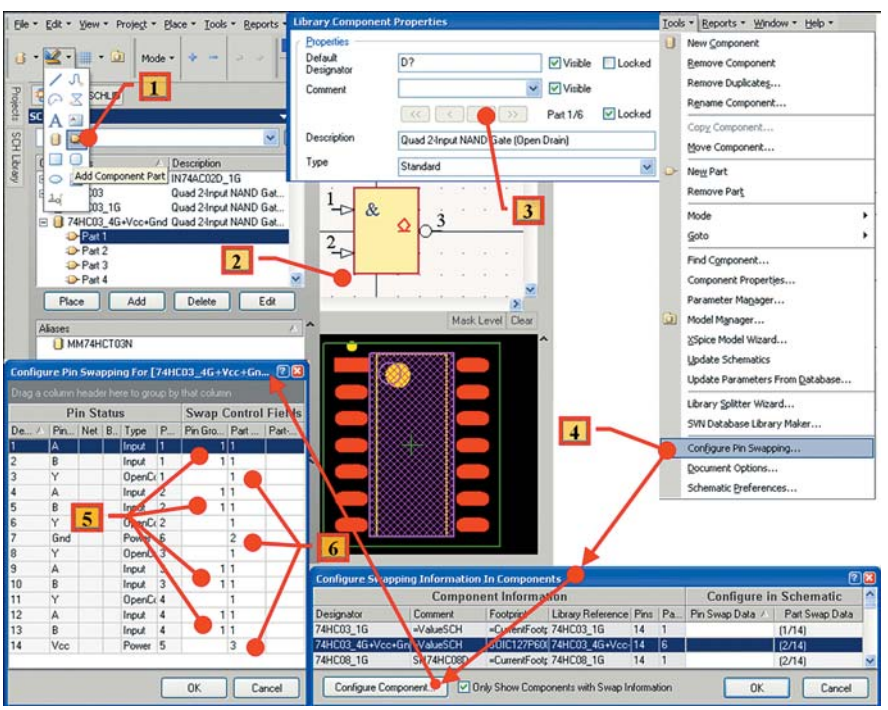

Рис. 37. Создание компонента с несколькими Part и настройка эквивалентности Pin и Part

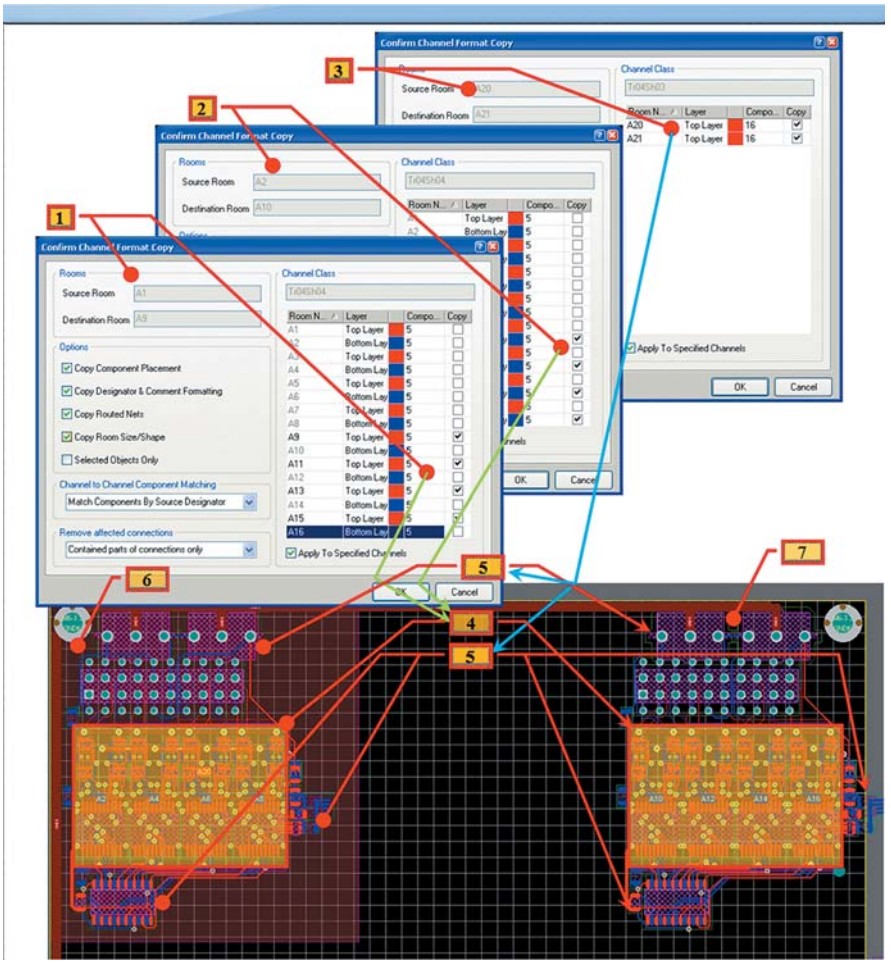

Рис. 38. Перенос топологии 8-канального АЦП

и **Pin=Gnd**) — уникальный (в нашем случае 2 и 3 соответственно).

Примечание. *Компоненты, содержащие информацию в столбцах* ***Swap Control Field****, можно группировать далее в библиотеках, схемах, PCB и других документах, формируемых командами меню* ***Report****.*

Аналогично можно поступить и с компонентом 74AC365.

Топология идентичных блоков модуля

Выделим и затем перенесем топологию 8-канального АЦП в левый угол печатной платы для соблюдения порядка нумерации каналов слева направо и доработаем топологию.

В прошлом номере мы рассмотрели вопросы создания топологии идентичных модулей. В данном примере мы имеем два идентичных модуля общих компонентов для двух 8-канальных сборок, содержащих 16 идентичных модулей каналов АЦП.

Аналогично предыдущему примеру перенесем топологию **Room** для четных и нечетных каналов АЦП (сначала только размеры и компоненты, затем, после расположения самих **Room**, и топологию). Точно так же поступим и с общей частью. Однако в этом случае следует размеры **Room A21** (общая часть первого модуля) сделать такими, чтобы внутри него были расположены **Room** с **A1** по **A8**. Примеры окончательных настроек при переносе топологии указаны на рис. 38, где отмечены:

1. Параметры переноса топологии канала АЦП № 1 на каналы № 9, 11, 13, 15.
2. Параметры переноса топологии канала АЦП № 2 на каналы № 10, 12, 14, 16.
3. Параметры переноса топологии **Room A20** (общая часть 1–8 каналов АЦП) на **Room A21** (общая часть 9–16 каналов АЦП).
4. Результат переноса топологии расположения каналов АЦП № 1–8 на каналы 9–16.
5. Результат переноса топологии общей части каналов АЦП № 1–8.
6. Топология 8-канального АЦП (каналы № 1–8) взята из предыдущего примера и расположена в данной области платы.
7. Топология 8-канального АЦП (каналы № 9–16).

После данных операций следует только откорректировать общие компоненты топологии 8-канальных АЦП с точки зрения зеркальной симметрии расположения соединителей и других элементов.

Рассмотрим топологию **Room A30**, в которой находятся компоненты с идентичными взаимозаменяемыми **Pin** и **Part**, в частности D6. В пакете проектирования **Altium Designer** предусмотрена возможность запрета или разрешения операции **Swap** (взаимной замены) идентичных **Pin** или **Part** для каждого компонента индивидуально. Эта функция удобна, например, для адресных шин, когда необходима не столько удобная топология, сколько регулярная последовательная структура компоновки электрических связей на выводах микросхемы. Однако и этот вопрос оставим для будущих примеров. В данном случае для компонента D6 разрешим взаимную замену идентичных элементов и сделаем операцию **Swap**. Для этого:

1. Откроем PCB проект (рис. 39).
2. Командой **Tool/Pin/Part Swapping/Configure…** откроем окно **Configure Swapping Information In Component**.
3. В данном окне для D6 разрешим взаимную замену идентичных **Pin** и **Part**.
4. Перейдем к компоненту D6. Для быстрого перехода к нему можно воспользоваться выпадающим меню.
5. В окне **Project/Project Option** разрешаем для данного проекта операции **Swap** для **Pin** и **Part**.

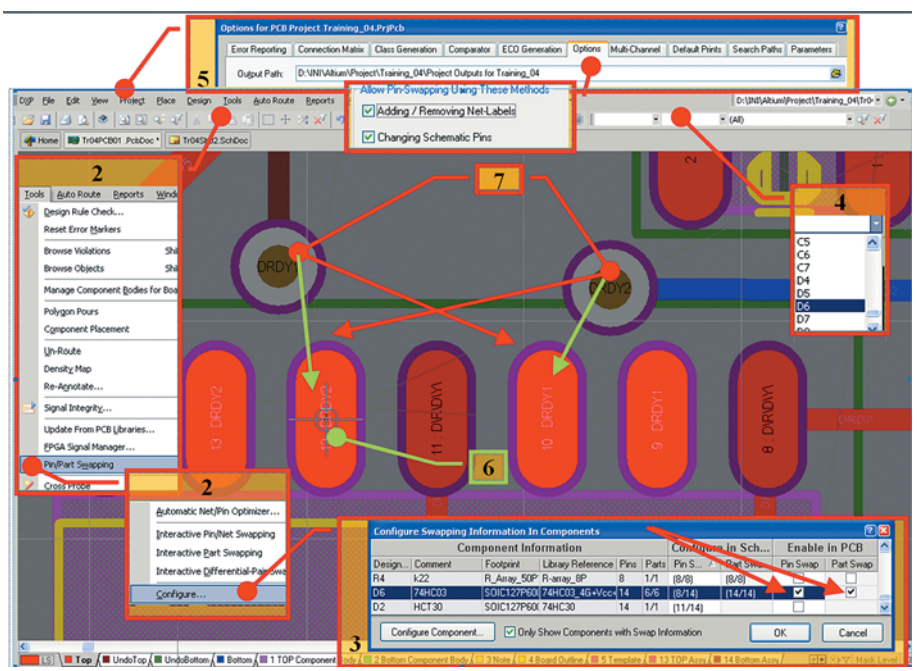

Рис. 39. Пример применения операции SWAP в PCB

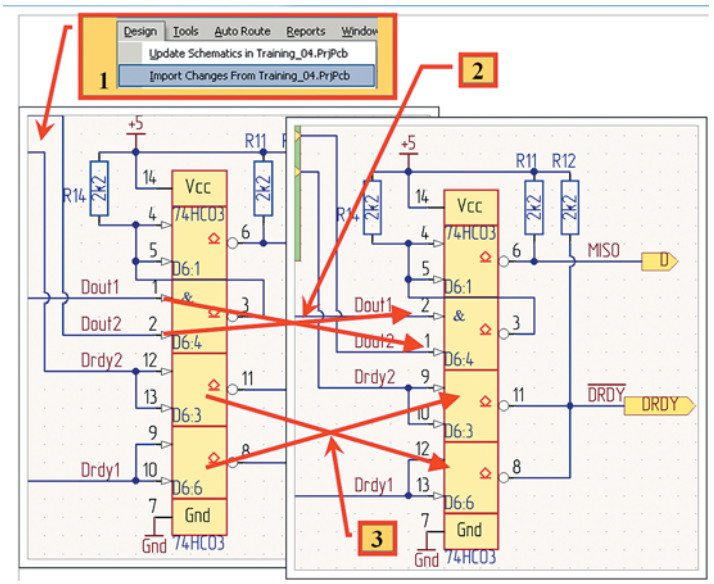

Рис. 40. Передача изменений в схему

6. Воспользуемся интерактивным способом замены, выполнив соответствующую команду из окна **Tool/Pin/Part Swapping**. При этом **Pin Part**, которым разрешена операция **Swap**, будут подсвечены более ярким цветом. При наведении указателя на один из **Pin** вокруг последнего возникнет восьмигранник, появление которого будет означать «захват» **Pin** для выбора. Выбираем сначала один из идентичных **Pin** первого **Part**, затем второго. Операция завершена.
7. Теперь топология в данном участке печатной платы более оптимальна (пересекающиеся связи, помеченные красными стрелками, заменены на непересекающиеся: они помечены зелеными стрелками).

Примечание. *Операция взаимной замены идентичных элементов компонента доступна только при отсутствии элементов топологии (дорожек и т. п.), подключенных к* ***Pin*** *этих элементов.*

Для примера таким же способом произведем взаимную замену в одном **Part**, идентичных **Pin** (в нашем случае входов логического вентиля), например, с номерами 1 и 2.

Аналогично можно поступить и с другими элементами, однако в данном примере мы этого делать не станем. Теперь следует передать изменения, сделанные для оптимизации топологии, в схему (рис. 40):

1. Из открытого проекта PCB командой **Design/ Update Schematics in…** загружаем изменения в схему.
2. Открываем лист схемы и убеждаемся, что **Pin** с обозначениями 1 и 2 действительно изменили свое положение.
3. Два идентичных **Part** также изменили свое положение в схеме.

Примечание. *Операция взаимной замены идентичных* ***Pin*** *элементов доступна только при отсутствии элементов топологии, подключенных к данному элементу.*

С точки зрения построения идентичных частей схемы 16-канального АЦП все завершено. На данном этапе мы не рассмотрели только общую часть электрической схемы и ее топологию. Добавим к проекту схемы микроконтроллера и интерфейсов. Поскольку практического интереса эти схемы не представляют, далее рассмотрим только особенности отдельных компонентов, используемых именно в них.

Реализация типовых схемных и топологических решений в примере

Обратимся к изображению компонента D3 (источник опорного напряжения) на рис. 41.

1. Как видно на рисунке, отображение вывода (**Pin 2**, **GND**) сделано не в соответствии с ГОСТ, согласно которому соответствующие надписи для данного вывода следует повернуть на 90°.
2. Открываем сначала библиотеку, а затем компонент MAX6129, соответствующий D3, и, наконец, открываем свойства для **Pin 2**.

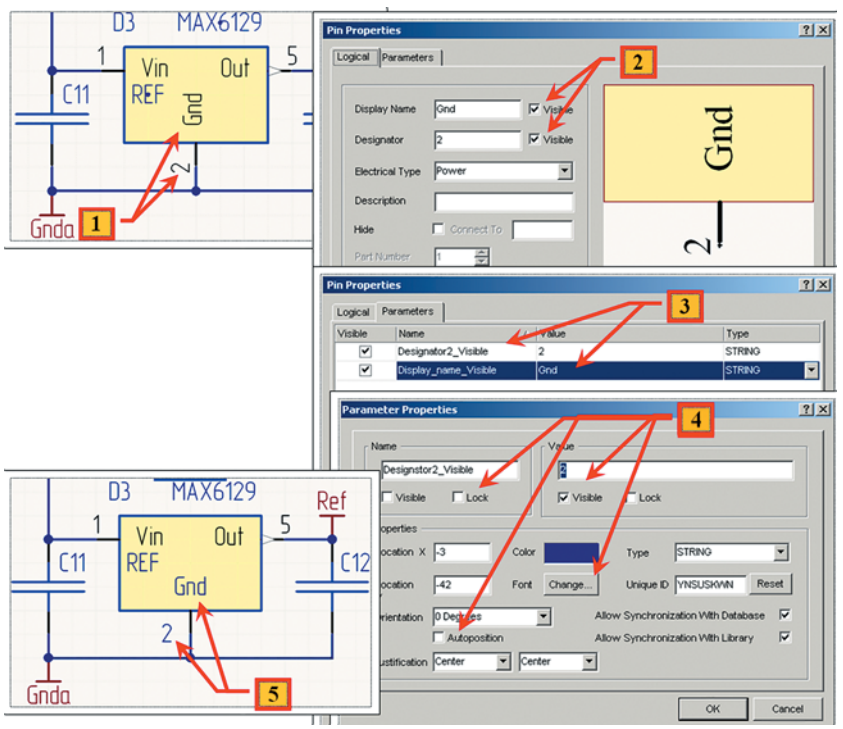

Рис. 41. Настройка отображения значений обозначения и названия Pin

В окне задания параметров для **Pin 2** на вкладке **Logical** снимаем флаг **Visible** отображения **Designstor** и **Display Name**.

3. Открываем вкладку **Parameter** и вводим два новых параметра, например:
   - **Designstor2_Visible** со значением, равным **2** (то есть равным значению **Designator**);
   - **Display_Name_Visible** со значением, равным **GND** (то есть равным значению **Display Name**).
4. Настраиваем вид отображения введенных параметров, в частности:
   - скрываем отображение названия параметра;
   - значение параметра делаем отображаемым;
   - указываем шрифт отображения значения параметра;
   - снимаем свойство **Autoposition**. Это необходимо, чтобы при перемещении компонента на схеме положение номера вывода и его название не изменяли бы свое положение относительно вывода **Pin**.
5. Новые значения для обозначения и названия **Pin** располагаем в удобном для прочтения месте или в соответствии с требованиями ГОСТ и обновляем компонент в схеме.

Примечание. *Следует учитывать, что связь между* ***Pin 2*** *компонента на схеме и соответствующим ему* ***PAD*** *посадочного места происходит по скрытому, а не по отображаемому на схеме номеру. Соответственно, существует опасность двух ошибок, связанных с таким вводом обозначений. Во-первых, ввод несоответствующего номера или названия; во-вторых, даже при правильном вводе номера или названия можно совместить видимое отображение этих параметров с другим* ***Pin****.*

На первых стадиях проектирования при первичной настройке модуля возможно к топологии вносить на печатную плату дополнительные перемычки или проводной монтаж.

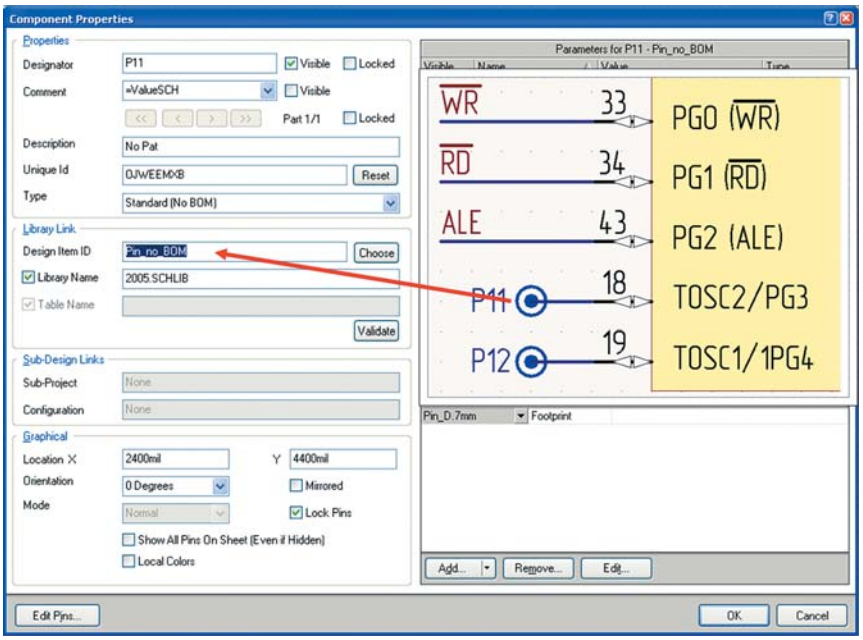

Рис. 42. Параметры компонентов, не входящих в перечень элементов

В этом случае желательно максимально возможное количество, а лучше все незадействованные выводы микросхем вывести на отдельные контактные площадки (более того, лучше, чтобы эти площадки были с переходным отверстием). Отличие таких компонентов на схеме от стандартных таково, что они имеют посадочное место, но не должны входить в перечень элементов. Это касается и специальных контрольных точек, используемых при наладке изделия или для контроля сигналов. Для того чтобы подобные элементы не входили в формируемый перечень, следует в их свойствах установить тип **Standart (No BOM)**, как показано на рис. 42.

Рассмотрим еще один специфический компонент, используемый в данном примере. Это топологическая перемычка. Она используется в двух местах: как средство для временного снятия или подачи определенных сигналов при настройке прибора и как способ задания адреса данного модуля. В последнем случае по умолчанию все перемычки установлены, а при окончательной сборке адрес модуля задается снятием соответствующей перемычки. При этом перемычки можно сделать спайкой рядом идущих проводников **Pad** данного компонента. Снятие перемычки — это снятие остатков припоя с проводников. На рис. 43 схематично представлен процесс создания (включая посадочное место) и применение такого компонента, где указаны:

1. Пример изображения компонента для схемы и часть схемы.
2. Положение посадочных мест на печатной плате.
3. Вид посадочного места в библиотеке.
4. **Pad** посадочного места. Вокруг **Pad** виден слой маски.
5. **Track** (линии топологии). Средняя соединена с одним **Pad**, остальные — со вторым.

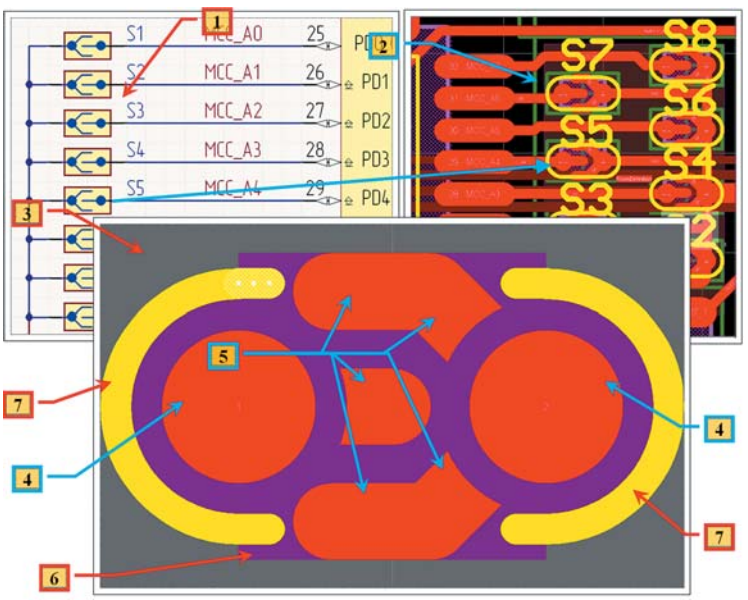

Рис. 43. Посадочное место с использованием Track

6. **Fill** — закрашенный прямоугольник в слое маски, предназначенный для открытия **Track** с целью дальнейшего нанесения припоя.
7. Нанесение изображения компонента.

Следующий сложный компонент — это использованные в примере соединители серии **FCI** (рис. 44). Особенность их в том, что они предназначены для подключения к ним штыревых соединителей как сверху, так и с обратной стороны платы. Более того, после окончательного монтажа эти соединители могут быть запаяны.

Особенность реализации компонента и его посадочного места заключается в следующем (пример приведен для 4-контактного соединителя):

1. Изображение компонента делаем обычным, с четырьмя **Pin**.
2. Контактные площадки **Pad**, предназначенные для контактов непосредственно соединителя, размещаемого на плате, нумеруем от 1 до 4.
3. Контактные площадки **Pad**, предназначенные для контактов непосредственно ответной части соединителя (штырей), также нумеруем — от 1 до 4. При этом у нас на посадочном месте будет два **Pad** с номером 1, два **Pad** с номером 2 и так далее.
4. Каждому **Pin** компонента будет соответствовать два **Pad** посадочного места, как показано для **Pad 1**.

При таком способе каждая пара **Pad** посадочного места будет соответствовать одному **Pin** компонента.

Использование полигонов

После завершения топологии электрических связей рассмотрим применение полигонов в данном проекте. Печатная плата здесь предназначена для компоновки в металлический корпус. Так как в данном примере мы использовали двустороннее расположение компонентов на плате и, более того, по возможности максимально — верхний и нижний слой для элементов топологии, размещение полигонов в данных слоях не дает никаких существенных преимуществ. Поэтому рассмотрим примеры размещения следующих полигонов:

- На слое **Top** и **Bottom** расположим по краю печатной платы полигоны для корпусной «земли», так как в этих местах печатная плата своим краем контактирует с металлическим корпусом.
- На внутренних слоях разместим полигоны для экранирования либо для усиления шин питания и «земли».

При создании полигона можно выбрать различные способы его заливки, в частности (рис. 45, ссылка 1):

- **Solid** — сплошная заливка полигона;
- **Hatched** — указание линий границ полигона и заливка внутренней области горизонтальными, вертикальными или теми и другими линиями с заданным шагом и как под прямым углом, так и под углом 45°. Окно для выбора последних настроек появляется только при использовании данного типа заливки полигона;
- **None** — без заливки полигона, а только с указанием его границы.

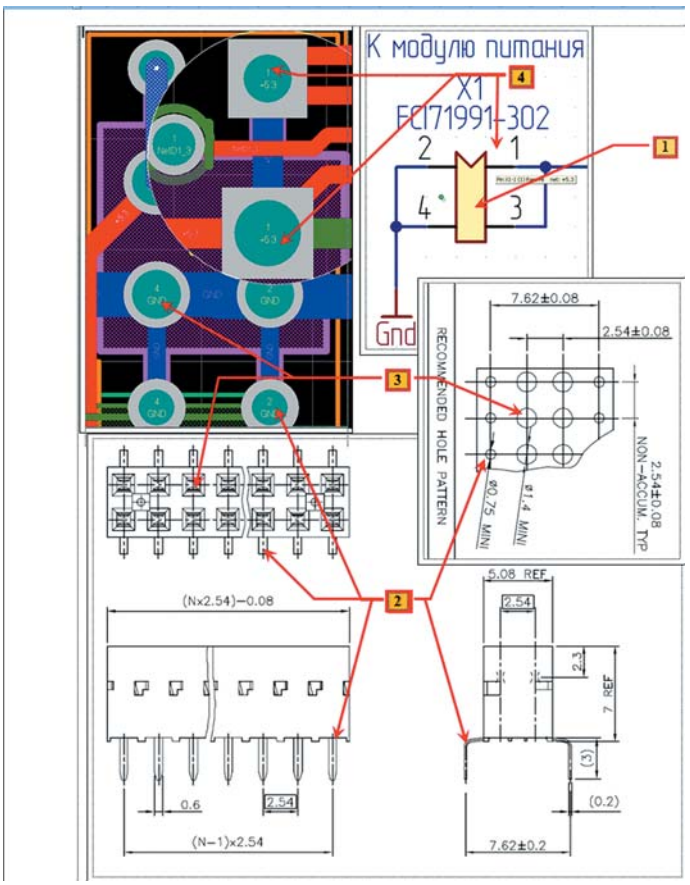

Рис. 44. Пример посадочного места с несколькими Pad для одного Pin

На рис. 45 представлен процесс создания полигона со сплошной заливкой на одном из слоев. На слое **Top** с помощью команды **Place/Polygon Pour** расположим прямоугольный полигон нужной ширины вдоль одной из сторон печатной платы. На рис. 45 (выноска 2) показаны настройки параметров полигона для слоя **Top**, где:

- **Layer => Top** — указан слой, где расположен полигон.
- **Name=> Gndk** — присвоено имя полигону. Имя может быть необходимо при написании правил. Если правила, использующие этот параметр, не будут применяться, — поле можно оставить незаполненным. Имя для правила мы выбрали так, чтобы оно совпадало с именем цепи электрической связи, к которой подключен полигон, хотя это и необязательно.
- Установлены (сняты) флаги:
  - **Lock Primitives** — запрет изменения формы полигона. Мы не будем пользоваться полигонами со сложным контуром, который необходимо редактировать.
  - **Locked** — защита полигона от выделения. Нам не нужно изменять положение полигона.
  - **Ignore On Line Violation** — используется при необходимости корректировки топологии для оптимальной заливки полигоном. При этом элементы топологии разрешено проводить через полигон, даже если он открыт. В нашем случае нам не требуется корректировка топологии.
  - **Remove Dead Copper** — удаление неподключенных «островов», образующихся при заливке полигона.
- **Connect to Net => Gndk** — указание электрической цепи, к которой будет подключен полигон, и способа заливки полигона:
  - **Pour Over All Same Net Objects** — закрытие полигоном элементов топологии (кроме **PAD**), в нашем случае принадлежащих **Gndk** (рис. 45, ссылка 3);
  - **Pour Over Same Net Polygons Only**;
  - **Don't Pour Over Same Net Objects** — все элементы топологии, кроме (в нашем случае) **PAD**, принадлежащих **Gndk** (рис. 45, ссылка 4, с указанием отличий от предыдущего варианта).

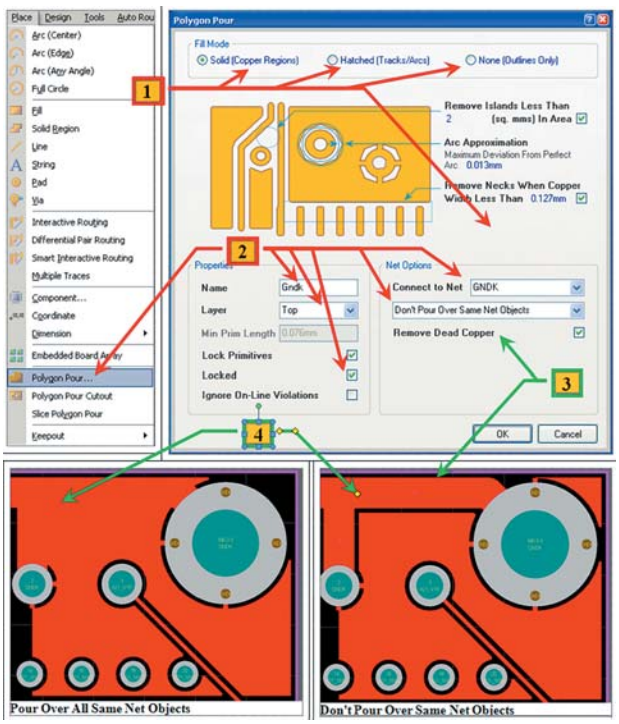

Рис. 45. Пример и вид настройки полигонов

Теперь разместим по сторонам платы четыре таких же полигона и еще четыре — с обратной стороны, для получения утолщенного контура меди по краям платы.

Однако для корпусной «земли» нас не устраивают ни значение величины зазора, ни ширина дорожек подключения полигона к **Pad**. Более того, при подключении полигона к **Pad** крепежного отверстия термобарьер вообще не нужен.

В настройках полигона зазоры между ним и элементами топологии не определены. Эти параметры устанавливаются с помощью правил, и сейчас мы рассмотрим некоторые из них.

С помощью команды **Design/Rules** вызываем окно задания правил (рис. 46, сноска 1). Переходим на вкладку **Polygon Connect Style** и добавляем новое правило **Polygon Connect** (рис. 46, сноска 1).

Создадим следующие правила для подключения полигонов **Gndk** к крепежным отверстиям (без термобарьера) и **Pad** токонесущих элементов (разрядников и элементов защиты с утолщенным подводом проводников при подключении полигона, для остальных элементов — с типовым подключением полигона):

- *Правило* ***PolygonConnect_GNDK-Direct*** для подключения крепежных отверстий (рис. 46).
  1. Вводим название и нажимаем кнопку **Query Builder** для вызова мастера построения правил.
  2. В окне мастера **Building Query Builder from Board** в первом столбце выбираем тип

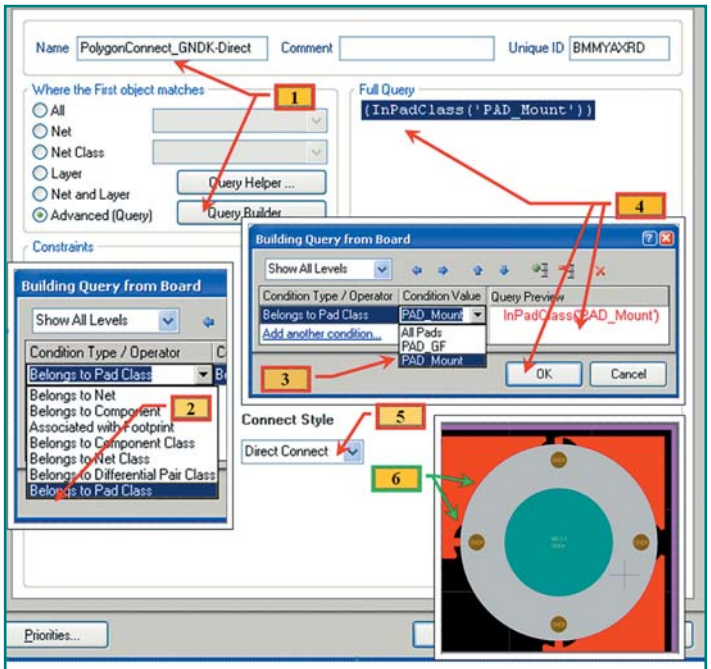

Рис. 46. Пример построения правила настройки подключения полигонов

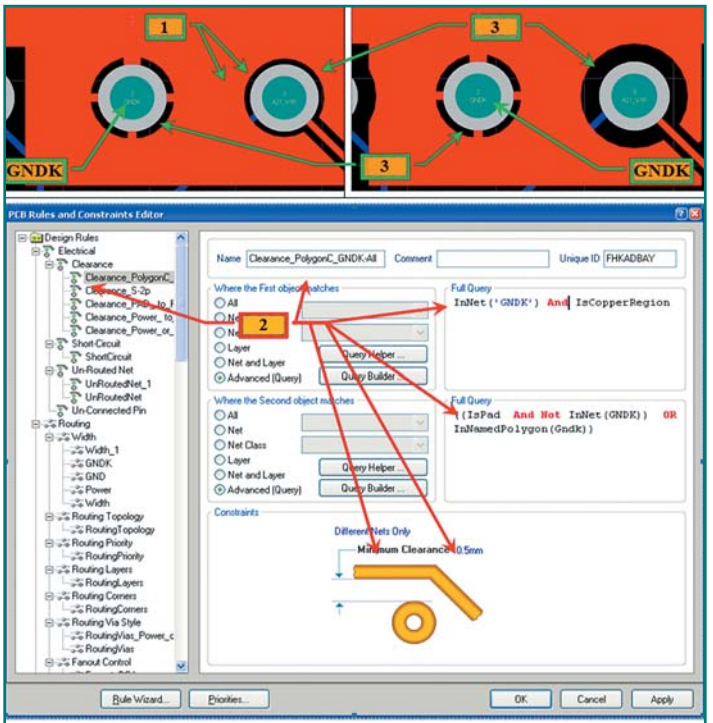

Рис. 48. Пример правил для обеспечения увеличенного зазора между элементами топологии

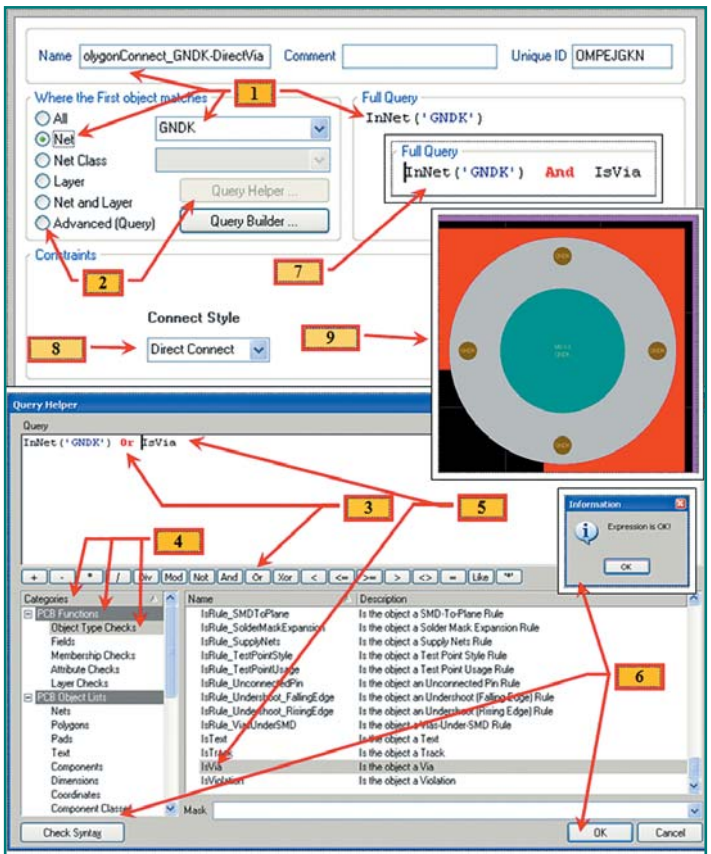

Рис. 47. Пример построения правил настройки подключения полигонов

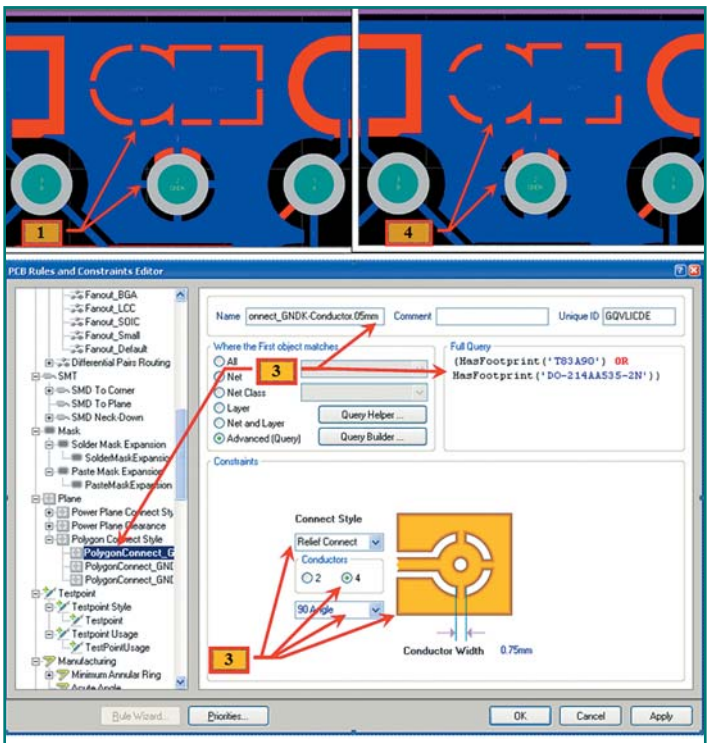

Рис. 49. Пример построения правил для задания ширины проводников подключения полигонов

объектов, на который будет действовать правило. В нашем случае это **Belong to Pad Class**, так как ранее для крепежных отверстий мы определили специальный класс.

3. Во втором столбце выбираем класс **Pad Mount**, которому принадлежат крепежные отверстия.
4. Соответствующий текст правила (**InPadClass ('PAD_Mount')**) отобразится в окне **Query Preview**, и после нажатия кнопки **OK** переместится в окно **Full Query** соответствующего правила.
5. И, наконец, вводим тип подключения (**Connect Style**) для этого правила как **Direct Connection**.
6. С помощью команды **Tools/Polygon Pour/ Repair All Polygon** производим перезаливку всех полигонов. В результате подключение к крепежному отверстию стало именно таким, как мы планировали. Однако форма крепежного отверстия у нас сложная, совмещенная с переходными отверстиями. И в их зоне имеются зазоры при подключении полигона.

- *Правило* ***PolygonConnect_GNDK-DirectVia*** для подключения полигона корпусной земли к переходным отверстиям (рис. 47).

1. Вводим название правила, указываем объект (**Net**), из выпадающего списка выбираем название **GNDK** и в окне **Full Query** получаем правило — **InNet ('GNDK')**. Однако таких условий недостаточно, так как в этом случае правило будет распространяться и на все **Via** и **Pad**. Нам для **Pad** следует оставить термобарьеры.

2. Указываем выбор объектов с помощью мастера (**Advance Query**), после чего кнопка **Query Helper** становится доступной. Нажимаем ее для вызова мастера построения правил.
3. В окне мастера **Query Helper** переходим в конец строки и нажимаем кнопку оператора And.
4. В столбце **Categories** выбираем **PCB Functions => Object Type Checks**.
5. В столбце **Name** выбираем **IsVia**, и соответствующая запись попадет в строку правил.
6. Нажимаем кнопку **Check Syntax** и проверяем синтаксис правила, после чего нажимаем кнопку **OK**.
7. Текст правила **InNet ('GNDK') And IsVia** переместится в окно **Full Query**.
8. Выбираем тип подключения (**Connect Style**) для этого правила — **Direct Connection**.
9. С помощью команды **Tools/Polygon Pour/Repair All Polygon** производим перезаливку всех полигонов. В результате подключение к крепежному **Pad** произойдет так, как нам необходимо.

- *Правило **Clearance_PolygonC_GNDK-All*** для подключения полигона корпусной земли к типовым **Pad** (рис. 48). Не будем подробно описывать их создание: процесс идентичен тем, что изложено в предыдущих пунктах. Отметим только характерные промежуточные этапы, отмеченные на рисунке.
  1. Недостаточно широкий зазор от полигона к элементам топологии, который не обеспечит нужную величину, как по запасу пробивного напряжения, так и по токам утечки.
  2. Вводим правило и указываем:
     a) тип (**Clearance**) правила;
     b) имя (**Clearance_PolygonC_GNDK-All**);
     c) первый тип объектов — **InNet ('GNDK') And IsCopperRegion**, означающий, что правило распространяется на все полигоны, подключенные к электрической цепи **GNDK**;
     d) второй тип объектов — ((**IsPad And Not InNet (GNDK)) OR InNamedPolygon (Gndk)**), означающий, что правило распространяется на все контактные площадки, не принадлежащие цепи **GNDK**, и элементы топологии (дорожки в виде линий и частей окружностей) внутри соответствующих полигонов;
     e) **Different Net Only** — указано, что правило действует на все цепи, кроме, в данном случае, **GNDK**;
     f) 0,5 мм — величина зазора для данного правила.
  3. Указан увеличенный зазор между контактными площадками, принадлежащей другой цепи, в соответствии с введенным правилом. При этом зазор термобарьера контактной площадки цепи **GNDK** остался прежним.
- *Правило **PolygonConnect_GNDK-Conductor.05mm*** для подключения полигона корпусной «земли» к контактным площадкам элементов защиты (рис. 49). Это правило необходимо для того, чтобы обеспечить максимальный ток, проходящий через контактную площадку, при сохранении термобарьера для обеспечения условий монтажа компонентов.
  1. Для данного случая рассмотрим полигон на обратной стороне печатной платы, так как именно там находятся контактные площадки всех компонентов. На рисунке отмечено место подключения полигона к контактным площадкам компонентов, предназначенных для большого импульсного тока. Ширина проводников при подключении указана по умолчанию, как для всех компонентов, и значение ширины следует увеличить.
  2. При создании нового правила указываем посадочные места (**HasFootprint('T83A90') OR HasFootprint('DO-214AA535-2N')**) компонентов, для которых следует провести такое подключение полигона.
  3. Выбираем стиль подключения полигона и его параметры (число, ширину проводников подключения и угол подвода к контактной площадке).
  4. В результате данный полигон будет подключен к контактным площадкам избранных посадочных мест с помощью более широких проводников.

Аналогично создаем правила и для остальных полигонов. Но отметим, что при необходимости «вырезов» в полигонах следует применить команду **Place/Polygon Pour Cutout**.

Итак, после завершения размещения полигонов все элементы топологии печатной платы в нашем проекте завершены. Однако перед формированием выходных файлов для производства печатной платы следует разместить все надписи на соответствующих слоях, предназначенных для производства.

Подготовка и размещение текстовой и пояснительной информации на печатной плате

Рассмотрим размещение и параметры обозначений компонентов на печатной плате.

Начнем с настройки слоев:

1. Командой **Design/Board Layer & Color** вызываем соответствующее окно и делаем следующие слои отображаемыми.
   – **Top** — для отображения всех **Pad** на верхней стороне;
   – **Top Solder** — для отображения слоя маски всех **Pad** на верхней стороне (надписи не должны совпадать с зонами очистки маски);
   – **Top Overlay** — слой, где будут нанесены надписи;
   – **1 TOP Component Body** — все надписи, которые должны быть видимыми после монтажа компонентов на печатную плату, наносятся с наружной стороны корпусов компонентов;
   – **Board Outline** — все надписи, отображаемые на печатной плате, должны быть нанесены внутри контура обрезки;
   – **13 TOP Assy** — для отображения вида корпуса посадочного места при размещении надписи.
2. Нажав на кнопку **LS** (**Manager Layer Sets**), вызываем меню, а командой **Board Layer Set** — непосредственно мастер настройки слоев. Этот пункт не обязателен, однако работа с данными слоями проводится практически во всех проектах, и желательно их однажды настроить и добавить в меню быстрого вызова (рис. 50).
3. Нажав на кнопку **New Set** и выбрав команду **From Current Visible Layer**, вводим новое имя для группы слоев.
4. Присваиваем имя новой группе, например **All Top Layer**, и указываем активный слой при открытии группы — **Top Overlay**.

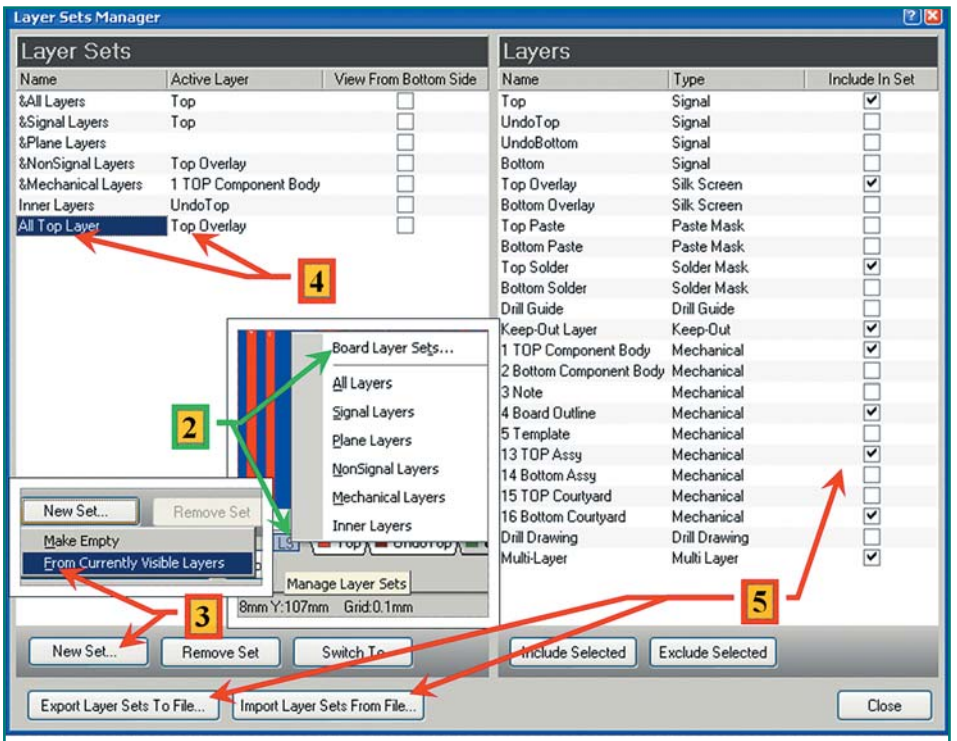

Рис. 50. Настройка группы слоев для удобства нанесения надписей

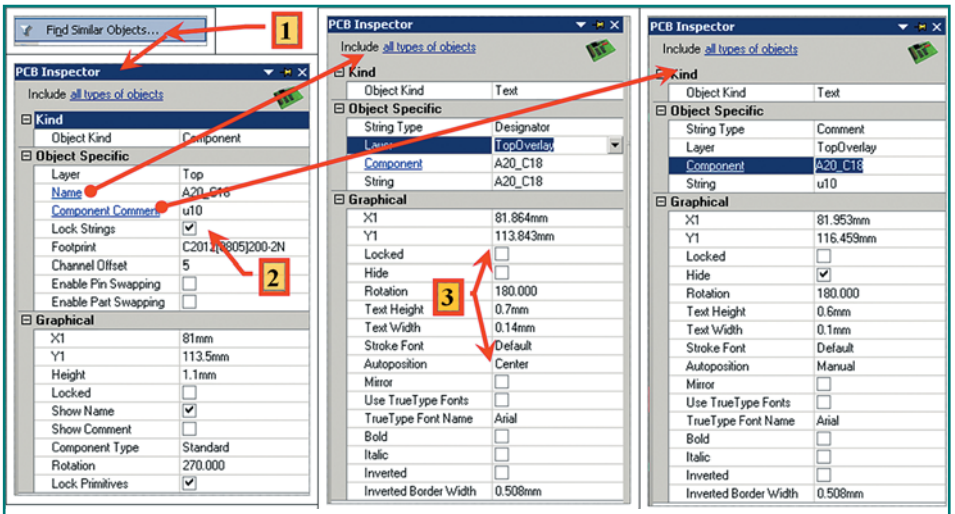

Рис. 51. Настройка параметров Designator и Comment

5. Вы всегда можете добавить и удалить слои из группы, а также сделать экспорт данных настроек слоев и, соответственно, импортировать их в другие проекты.

Аналогично настроим новую группу слоев и для нижней стороны печатной платы.

Данный проект мы разрабатывали, используя старые и создавая новые компоненты, и поэтому все надписи у компонентов могут иметь разные значения, как по размеру текста, так и по его положению. Поэтому следует вначале привести все надписи к единому виду и уже на месте изменять их параметры и положение.

1. Выделим все компоненты и вызовем инспектор (клавиша **F11**).
2. Снимем флаг **Lock String**, чтобы иметь возможность изменить место расположения надписи относительно посадочного места, и затем нажмем ссылку **Name** вызова окна параметров **Designator** компонентов.
3. Для всех компонентов установим следующие параметры текста **Designator**:
   - **Locked** — флаг снят (для возможности изменения положения и «захвата» при выделении);
   - **Hide** — флаг снят (для отображения параметра);
   - **Rotation** — 0.000 (для отображения текста параметра слева направо);
   - **Text Height** — высота текста. Менее 0,5 мм не стоит использовать, так как текст не будет различим. Для данной плотности компонентов на печатной плате и их размеров можно выбрать высоту равной 1 мм;
   - **Text Width** — ширина линий текста. Менее 0,1 мм не стоит использовать, так как текст будет не различим. Для нашего примера укажем ширину текста 0,15 мм;
   - **Stroke Font** — **Default**. Один из доступных типов шрифта;
   - **Autoposition** — **Center**. Все **Designator** разместить в центре соответствующего компонента (так легче при первой расстановке видеть соответствие **Designator** своему компоненту);
   - **Mirror** — оставить все, как есть, но на следующем этапе при расстановке элементов на нижней стороне платы следует активизировать это свойство;
   - **Use True Type Font** — флаг снят. Применение данного типа шрифтов приводит к появлению множества полигонов, используемых для отображения букв в надписях, и, соответственно, к значительному увеличению размера **Gerber**;
   - Остальные параметры для **Designator** не стоит изменять относительно значений, установленных по умолчанию.
4. Аналогично поступаем с параметрами **Comment** компонента за исключением (рис. 51):
   - **Locked** — флаг установлен. Как правило, этот параметр нужен только для сборочных чертежей, и в его перемещении нет необходимости.
   - **Hide** — флаг установлен (для скрытия параметра. На данном этапе он нам не нужен).
   - **Rotation** — не изменяем значение.
   - **Text Height**, **Text Width** — как правило, размер можно установить больше, чем для **Designator**, так как места для их размещения на печатной плате больше.
5. **Use True Type Font** — флаг можно установить, при этом надписи в сборочных чертежах будет легче читать.

Теперь откроем проект и расставим на верхней стороне печатной платы все надписи типа **Designator** (не забывайте использовать опцию переноса для идентичных **Room**), а также вспомогательные надписи, например, как показано на рис. 52, где отмечены:

1. **Designator** конденсатора C19.
2. Информационная надпись имени файла печатной платы — для быстрого поиска (при необходимости) исходных файлов.
3. Номер разработки и версии топологии — для поиска технической документации и регистрации и идентификации, например, у производителя фотошаблонов.
4. Контур расположения одного канала.
5. Информационная надпись имени канала АЦП (**Room**) — для определения местоположения данных каналов.
6. Место для нанесения индивидуального номера платы. Операция выполнена с применением инверсного **True Type** шрифта.
7. Информационные надписи для указания выходных сигналов на соединителе.

Для нижней стороны печатной платы все надписи должны иметь свойство **Mirror**. Однако расстановка надписей с зеркальным отображением вызывает затруднение из-за сложности их прочтения. Поэтому перед расстановкой надписей с нижней стороны печатной платы следует воспользоваться опцией **View/Flip Board**. В этом случае вид платы «перевернется», верхняя и нижняя сторона поменяются местами, и все надписи с нижней стороны станут легко читаемы, несмотря на опцию **Mirror**.

В окончании статьи мы рассмотрим вопросы подготовки технической документации и файлов для производства и монтажа печатных плат.

Окончание следует

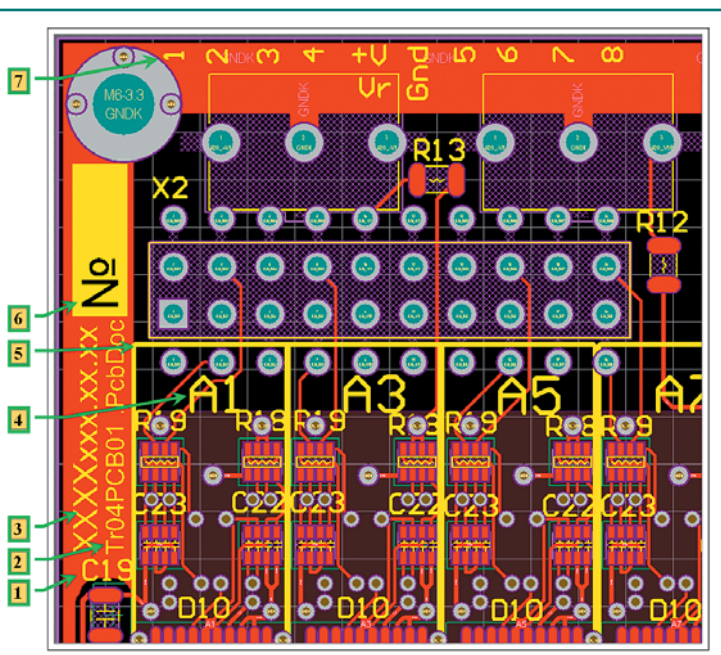

Рис. 52. Пример нанесения маркировки