

Altium Designer 6 в примерах

В данной статье будет представлен новый пример схемной и топологической реализации многоканального модуля ввода/вывода.

Владимир Пранович,
к. т. н.

pranovich@bsu.by

Рассмотрим следующий проект, содержащий не много компонентов, однако достаточно сложный как в реализации его схемы, так и с точки зрения топологических решений печатной платы.

Проект содержит схему на основе контроллера DS31256, с помощью которого реализован мост между PCI-шиной и 16 каналами ввода/вывода E1/T1. Последние в свою очередь реализованы на основе 4-канальных трансиверов DS21Q50LN и включают согласующие цепи.

Не будем останавливаться на приемах создания электрической схемы, так как о них подробно рассказано в статье [1]. Укажем только основные фрагменты схемы и те методы, которые использовались при их создании, но ранее детально не были описаны.

Параметры проекта

Схема данного проекта располагается на нескольких листах и имеет иерархическую структуру. Она содержит общую часть, а также схему для 16 идентичных блоков каналов ввода/вывода. Поэтому командой **Project\Project Option\Option** (рис. 1) вызываем окно параметров и устанавливаем там следующие свойства:

1. В строке **Output Patch** указываем путь к папке, где будут храниться вспомогательные и выходные файлы, генерируемые пакетом проектирования.

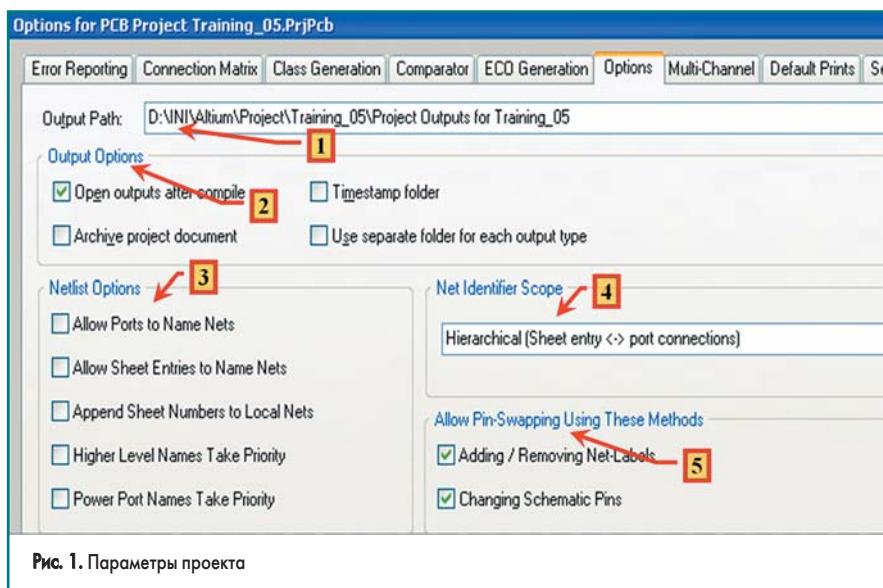


Рис. 1. Параметры проекта

2. В зоне **Output Option** установка или снятие флагов для данного примера не оказывают никакого влияния.
3. В зоне **Netlist Option** снимаем все флаги. Все имена цепей будем задавать через метки **NetName**.
4. В зоне **Net Identifier Scope** выберем из выпадающего списка значение **Hierarchical (Sheet Entry <-> port connection)**, тем самым определив иерархическую структуру проекта и тип межлистовых связей — через **Sheet Entry** и соответствующий ему **Port** подчиненного листа.
5. В зоне **Allow Pin Swapping Using These Methods** установим все флаги. Хотя для данного примера это не принципиально, так как у нас не будет компонентов, для которых возможна взаимная замена идентичных выводов или частей компонента.

Первый лист схемы

Он содержит все конденсаторы, предназначенные для фильтрации, а также компоненты, в состав которых входят в основном только выводы шин питания или специфические выводы, не имеющие прямого отношения к схемной реализации самого устройства ввода/вывода.

На рис. 2 представлена часть первого листа схемы. На рисунке указаны:

1. Ссылки (**Sheet Entry**) на второй (с обозначением листа **A6**) и третий (с обозначением **A7**) лист электрической схемы проекта. Заметим, что обозначение **Sheet Entry** лучше всегда изменить со значения, присваиваемого по умолчанию, на другое. Например, применить последовательную нумерацию. В данном примере для обозначения используется буква «А» и цифра, но поскольку эта схема — часть более сложного проекта, автор не восстанавливал последовательную нумерацию.
2. Первая часть (**Part**) компонента DS31256. Она содержит служебные выводы контроллера, а на схеме указано подключение этих выводов для данного проекта.
3. Пятая часть (**Part**) компонента DS31256. Она содержит все выводы питания и «земли» контроллера. Соответственно на схеме указаны подключения данных выводов.
4. Третья часть (**Part**) компонента топологического соединителя для шины PCI. В данной части сведены все выводы (**PIN**) соединителя, относящиеся к шинам питания и «земли», а также другие

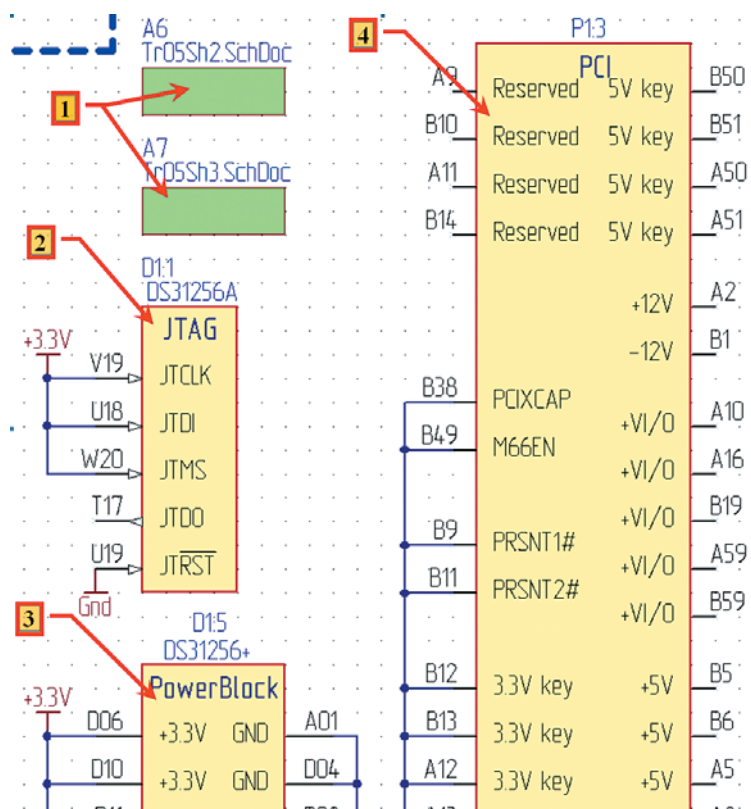


Рис. 2. Фрагмент первого листа схемы

Таким образом, на первом листе у нас будет размещена часть схемы, описывающая подключение питания шины PCI и микроконтроллера.

Второй лист схемы

Здесь разместим все элементы, имеющие отношение к электрическим цепям шины PCI. На рис. 3 указаны:

1. Первая часть (**Part**) компонента топологического соединителя для шины PCI. В ней расположены все выводы (**Pin**) соединителя, не относящиеся к шинам питания и адресным сигналам.
2. Третья часть (**Part**) компонента DS31256. В ней расположены выводы контроллера, предназначенные для связи с шиной PCI.
3. Вторая часть (**Part**) компонента топологического соединителя для шины PCI, которая содержит только адресные выводы.

Таким образом, на втором листе будет размещена часть схемы, описывающая подключение шины PCI к микроконтроллеру.

Такое решение схем первого и второго листа позволяет использовать их и для других проектов, так как эти схемы описывают типовое назначение сигналов шины PCI.

Третий лист схемы

Он будет содержать схему генератора, а также общую часть контроллера, связанную с трансиверами, и ссылки на их схемы. Рассмотрим эту часть подробнее.

На рис. 4 показан лист схемы в целом и наиболее информативные части в увеличенном виде:

1. Четвертая часть (**Part**) компонента DS31256. В ней расположены выводы контроллера, необходимые для управления индивидуально каждым из 16 каналов цифрового ввода/вывода. Схема не представляет большого интереса, ее прототип описан в документе DS31256DK.pdf, который можно найти на сайте производителя. Укажем только, что в данном проекте сигналы TC0...TC7 и TC8...TC15 объединены и соединены с выходом генератора 2,048 МГц. Такое соединение по группам вызвано топологическим расположением трансиверов попарно, с размещением генератора между парами трансиверов. Выход генератора нагружен на большое число входов, и более того, вследствие специфики этого сигнала относящиеся к нему топологические элементы требуют особого внимания.
2. Вторая часть (**Part**) компонента DS31256. В ней расположены адресная шина, шина данных и другие выводы — для общих сигналов, необходимых при обслуживании всех каналов ввода/вывода. Все неиспользованные выводы подключены к контрольным площадкам. Это сделано для возможности использования и этих выводов при настройке или модернизации устройства.
3. Обратите внимание: на примере сигнала **WR** показано, что общие сигналы для всей схемы целиком (а также сигналы **RD** и **INT**)

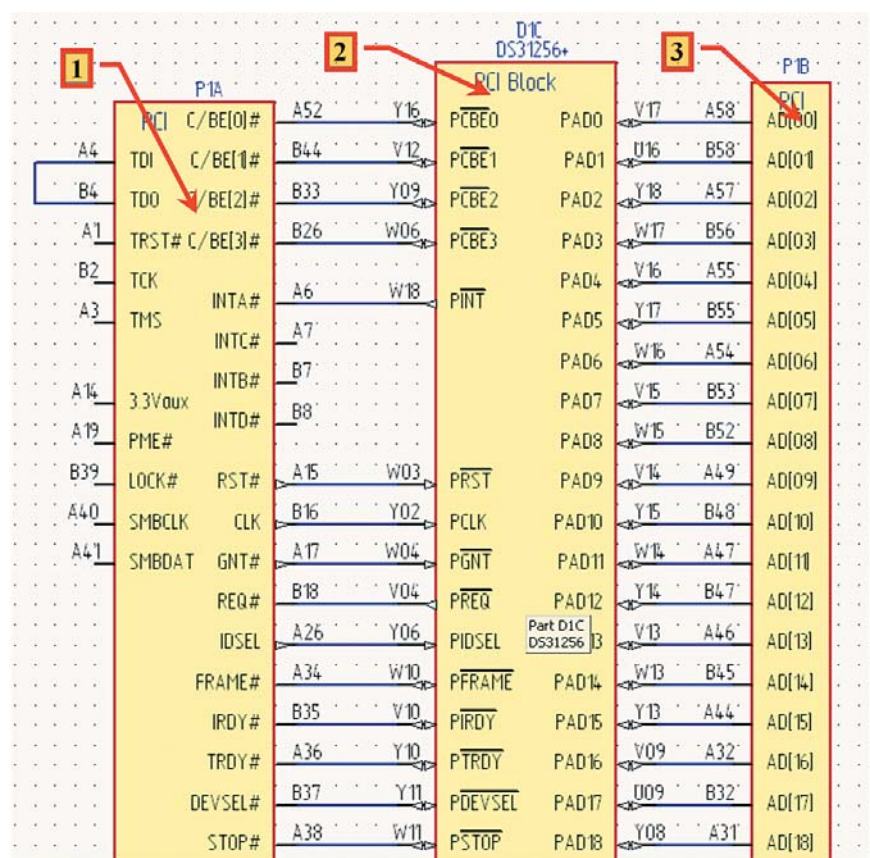


Рис. 3. Фрагмент второго листа схемы

служебные выводы, которые не используются непосредственно микроконтроллером. Особенности этого компонента рассмотрим более подробно ниже, так как здесь применяются интересные подходы при создании

как схемного изображения компонента, так и посадочного места. Более того, придется произвести дополнительные действия с этим компонентом уже непосредственно в редакторе топологии платы.

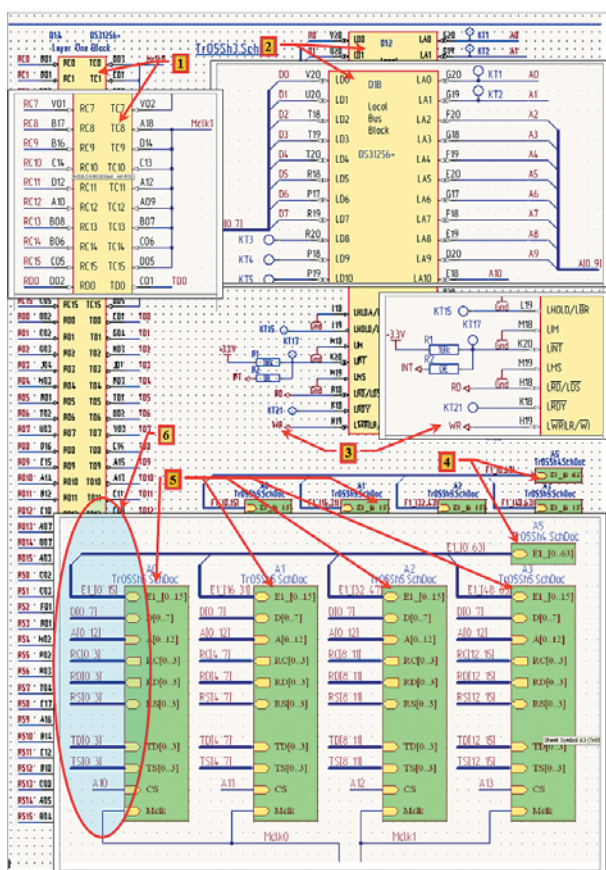


Рис. 4. Третий лист схемы

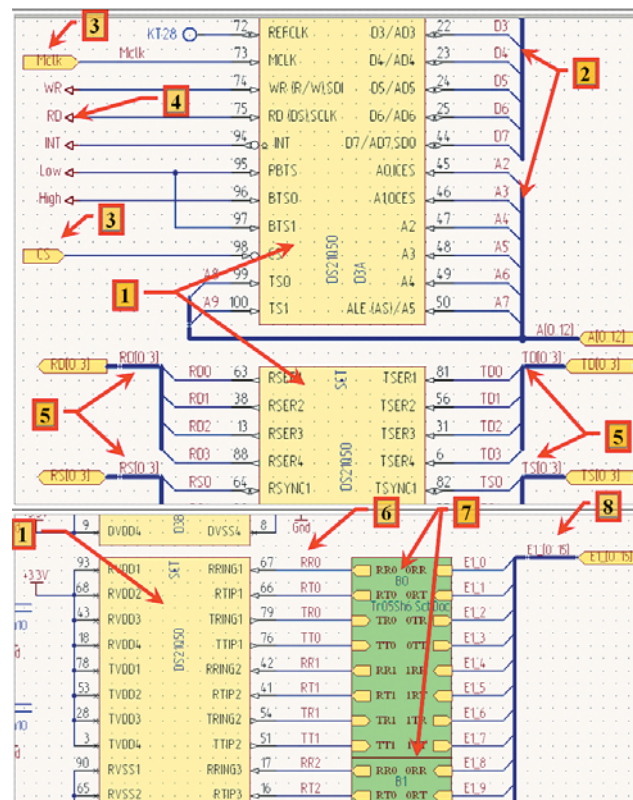


Рис. 5. Пятый лист схемы

будут связываться через порты **Power Port**. Таким образом можно передавать общие для всех листов сигналы. Заметим, что в версии Altium Designer 6.8 появились новые средства для передачи связей между разными листами проекта. Однако об этих преимуществах мы расскажем в отдельной статье.

4. Ссылка (**Sheet Symbol** с обозначением **A5**) на лист, где будет находиться изображение только выходного соединителя, для подключения всех внешних сигналов ввода/вывода. Изображение выходного соединителя на одном листе удобно как с точки зрения быстрого нахождения выходных сигналов, так и тем, что при многократном использовании подчиненного листа все элементы в нем должны иметь индивидуальные посадочные места. В нашем примере с одним соединителем на четыре идентичных модуля мы не можем разбить его изображение на отдельные части (**Part**) и использовать их индивидуально в повторяющихся блоках. Итак, в нашем примере данный **Sheet Symbol** содержит только один **Sheet Entry** с именем **E1_[0...63]** на 64 входа (32 пары для 16 входных сигналов каналов ввода **E1** и 32 пары для 16 выходных сигналов каналов вывода **E1**). Соответственно ввод электрических сигналов производится через шину **NetName=E1_[0...63]**.

5. Наиболее интересная часть листа этой схемы. Здесь указаны четыре **Sheet Symbol** (**A1–A4**) со ссылкой на один и тот же лист схемы трансивера. В данном примере выбран способ построения схемы с четырьмя идентичными блоками. По мнению автора, в этом

случае, во-первых, схема несет большую информационную нагрузку, чем при использовании одного **Sheet Symbol** со значением **Designator = (A1,4)** (такой способ мы применяли в предыдущем примере), во-вторых, применение последнего способа невозможно для более ранних версий (6.7 и ниже) Altium Designer 6. В дальнейшем мы покажем принципиально новые возможности последнего (6.8) обновления Altium Designer именно в реализации иных способов использования межлистовых связей.

6. Теперь рассмотрим способ межлистовых соединений, примененный в данном примере. Итак, все четыре **Sheet Symbol** идентичны, включая расположение **Sheet Entry**, за исключением индивидуальных **Designator**. Однако при подключении к ним шин и электрических связей используется следующий метод:

- Входы/выходы сигналов ввода/вывода данных (**Sheet Entry=E1_[0...15]**) первого, второго, третьего и четвертого **Sheet Symbol** подключены соответственно к сигналам **E1_[0...15]**, **E1_[16...31]**, **E1_[32...47]**, **E1_[48...63]** и поступают на лист (**Sheet Symbol = A5**);
- аналогично созданы все индивидуальные цифровые управляющие сигналы (**RC[0...15]**, **RD[0...15]**, **RS[0...15]**, **TD[0...15]**, **TS[0...15]**), связанные с микропроцессором, только они группируются соответственно по четыре на каждый **Sheet Symbol** (для первого с номерами **[0...3]**, второго — с **[4...7]**, третьего — с **[8...11]**, а четвертого — с **[12...15]**);

- общие сигналы (шина данных **D[0...7]**, адресная шина **A[0...12]**) поступают на все четыре **Sheet Symbol**;
- старшие адреса **A10**, **A11**, **A12**, **A13** дополнительно поступают соответственно на **Sheet Entry = Cs** и образуют, таким образом, сигналы выбора соответствующего трансивера, с которым производится обмен по шине данных и адресной шине. Обратите особое внимание на то обстоятельство, что в подчиненном листе один и тот же сигнал может поступать через разные порты ввода и, соответственно, иметь различное назначение в подчиненной схеме;
- и, наконец, сигналы **Mclk** поступают на две пары трансиверов для объединения их только на генераторе, как мы уже упоминали выше. Схема генератора на рисунке скрыта наложенным изображением.

Четвертый и пятый листы схемы

На четвертом листе расположен только один элемент — внешний соединитель, для ввода/вывода сигналов, и мы его приводить не будем. Перейдем сразу к пятому листу, где находится схема 4-канального трансивера, состоящая из трех частей: непосредственно микросхемы; фильтрующих конденсаторов; ссылок на согласующую часть каналов ввода/вывода. Характерные фрагменты схемы приведены на рис. 5, где указаны:

- Три части (**Part**) трансивера:
 - первая часть (сверху) — там совмещены все выводы трансивера, общие для всех

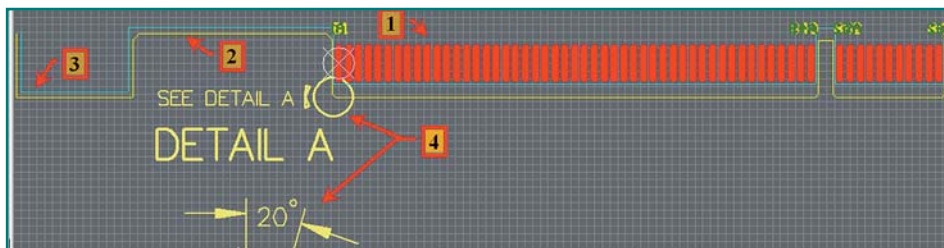


Рис. 8. Посадочное место PCI

6. **Dimension** (указание основных размеров). Выполнено в слое **Mechanical4**. Удобно использовать для контроля всех размеров, однако эти элементы не могут быть перенесены в библиотечный компонент.

7. Крепежное отверстие.
8. Технологическая надпись и ссылка на чертеж.
9. Непосредственно чертеж с указанием вида и параметров механической обработки контура, со стороны ламелей посадочного места.

Для того чтобы «вытащить» посадочное место слота PCI из проекта, проведем следующие операции (подробно они описаны в [1]):

1. Откроем указанный документ.
2. Командой **Design\Make PCB Library** создадим библиотеку посадочных мест.
3. В библиотеке **PCI short card 3.3V — 32 BIT.PCBLib** будет сформирован единственный компонент с именем **PCI3V32BIT**. Откроем панель **PCB Library**, выделим данный компонент и скопируем его в буфер.
4. Откроем нашу библиотеку посадочных мест. Откроем панель **PCB Library** и вставим новый элемент из буфера.

Теперь в нашей библиотеке есть прототип будущего посадочного места. Однако в нем отсутствуют следующие элементы:

- контур нижней стороны платы;
- указание на положение передней стороны платы;
- указание на минимальное положение задней стороны платы;
- элементы чертежей с правилами механической обработки платы со стороны ламелей;
- зоны запрета трассировки в местах глубоких вырезов посадочного места PCI и крепления (по крайней мере, в нижней части) планки соединителя.

Можно скопировать указанные объекты из файла **PCI short card 3.3V — 32 BIT.PCBDOC**

и перенести их через буфер в библиотечный компонент. Однако, во-первых, наш проект будет размещен на плате еще меньшего размера (по ширине печатной платы ограничимся минимальным размером, по высоте определимся в процессе создания топологии), во-вторых, создавать сложные контуры проще непосредственно в редакторе топологии, а не в редакторе посадочных мест, так как последний не поддерживает **Dimension**. Изменять положение линий контура и отслеживать их расположение друг относительно друга значительно проще, если данные объекты связаны через **Dimension**. Поэтому вернемся в файл топологии (**PCI short card 3.3V — 32 BIT.PCBDOC**) и откорректируем указанные объекты в исходном файле. Более того, перенесем при необходимости их изображения на слои, назначенные для этих объектов в нашей библиотеке. Так, для контура печатной платы в предыдущих примерах мы использовали слой **Mechanical4**, переименованный в **Board Outline**, все информационные надписи и чертежи — в слоях **Note** и **Dimension**.

После указанных переделок в файле печатной платы скопируем необходимые элементы и через буфер обмена перенесем в нашу библиотеку для посадочного места **PCI3V32BIT**. Будьте внимательны и при копировании не забудьте указать точку привязки, например центр первой контактной площадки. Так вам будет легче совместить объекты при вставке из буфера.

В результате получаем посадочное место с элементами контура, зон запрета трассировки и др. (рис. 8), где указаны:

1. Контактные площадки и другие элементы посадочного места, сформированные из прототипа. На первом контакте указана точка привязки посадочного места (**Edit >> Origin >> Set**).

2. Контур запрета топологии (добавлен в компонент).
3. Контур печатной платы (добавлен в компонент).
4. Чертеж, с требованиями механической обработки (добавлен в компонент копированием из прототипа печатной платы).

Создание шаблона печатной платы

В качестве основы возьмем файл предыдущего примера. Однако вначале сделаем из него шаблон, для того чтобы в дальнейшем не повторять эти операции для новых проектов. Для этого:

1. Скопируем файл **Tr04PCB01.PcbDoc** предыдущего примера, например в папку **.../Template**, где мы храним шаблоны, с именем **4LaySPSS.PcbDoc** (четыре слоя, сигнальный, слой типа **Plane** и еще два сигнальных).
2. Удалим все посадочные места, контур печатной платы и другие элементы, имеющие отношение исключительно к прототипу.
3. Добавим простой прямоугольный контур платы и контур в слое **Keep-Out Layer**.
4. Удалим все специфические классы цепей и правила, оставив только типовые, например для цепей питания (**Design\Rules, Design\Classes**).
5. Командой **Design\Layer Stack Manager** вызываем окно настройки параметров физических слоев шаблона (рис. 9). Аналогично предыдущему примеру [2007, № 5], создадим стек из четырех слоев, только слою **Undotop** присвоим тип **Plane** и назначим его электрической связи **GND**, как указано на рис. 9.

Создание контура печатной платы

На сайте производителя находится описание платы с установленным на ней контроллером DS31256, которая специально разработана для построения прототипов схемы. С учетом того, что на плате есть монтажное поле, где находятся контактные площадки для всех сигналов контроллера, плата имеет шесть слоев. Мы ограничимся стеком из четырех слоев. При возникновении сложности топологического решения для нашего проекта добавим в стек дополнительные слои.

Для создания контура печатной платы нам понадобятся только что созданное посадочное место, а также шаблон для нового файла **4LaySPSS.PcbDoc** — файла топологии.

Произведем следующие операции (рис. 10):

1. Скопируем шаблон в папку нашего проекта и переименуем его (в нашем случае — на имя **TR05.PcbDoc**).
2. Используя команду **Project\Add Existing to Project**, добавим к нашему проекту файл **TR05.PcbDoc**.
3. Теперь осуществим синхронизацию схемы и файла топологии, используя команду **Design\Import Changes From Training_05.PrjPcb**.
4. Переместим посадочное место слота PCI (в нашем случае — компонент P1) на место, где удобно расположить контур будущей платы.

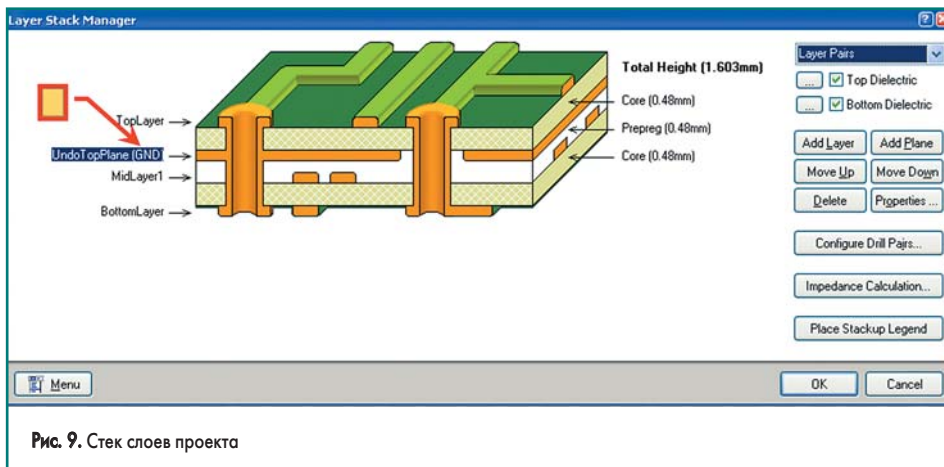


Рис. 9. Стек слоев проекта

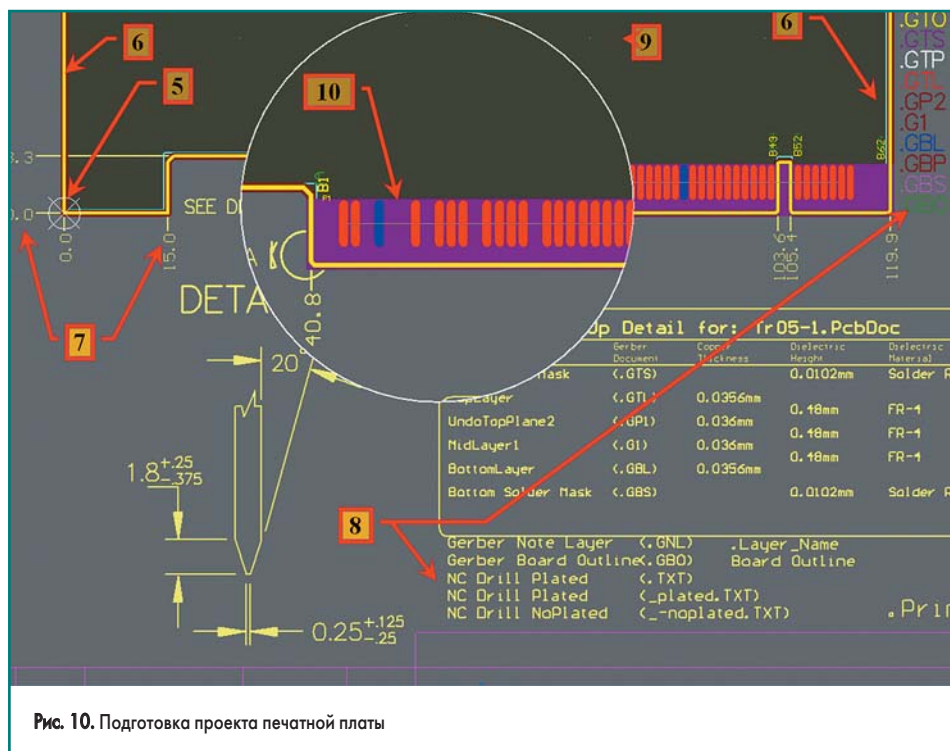


Рис. 10. Подготовка проекта печатной платы

5. Установим точку привязки (отсчета координат) в левый нижний угол начала компонента (посадочного места) при помощи команды **Edit\Origin\Set**.

6. Удалим контур платы и контур запрета трассировки, оставшиеся от шаблона, и добавим свои для данного примера. Учтите: с нижнего края печатной платы они уже созданы в посадочном месте, и можно добавить только недостающие стороны!

7. Добавляем в слое **Dimension** характерные размеры.

8. Располагаем остальные элементы служебной информации в свободных местах, таким же образом, как мы делали в предыдущем примере.

Примечание. При изменении **Layer Stack** следует все надписи, относящиеся к нему, удалить и восстановить вновь командой **Tool\Layer Stack Legend**.

9. Выделяем все элементы на слое **Board Outline** и командой **Design\Board Shape\Define from selected object** создаем область печатной платы.

10. В нашем проекте используются не все выводы компонента слота PCI. Так как на контактные площадки его посадочного места должно наноситься золотосодержащее покрытие, для экономии драгметаллов можно скрыть незадействованные контактные площадки.

Скрытие неиспользуемых контактных площадок в посадочном месте

Самый простой и быстрый способ — выделить данные контактные площадки и удалить. Однако у него есть существенный недостаток — если в процессе работы над проектом придется задействовать один из таких выводов, сделать это можно будет только через удаление посадочного места и восстановление его в проекте из библиотеки. Поэтому предложим

альтернативный способ — скрыть неиспользуемые выводы. Для этого произведем следующие операции (рис. 11):

1. Выделим в посадочном месте контактную площадку, которая не имеет электрических связей, и, нажав на правую кнопку мышки, выделим операцию **Find Similar Object**.

2. В окне **Find Similar Object** добавим условие: **Net=No Net (Same)**, то есть выделить все контактные площадки, не имеющие электрических подключений.

3. Нажатием клавиши **F11** вызываем инспектор и затем изменяем размеры выделенных контактных площадок на нулевые значения, а также указываем признак полного закрытия их маской.

После осуществления этих действий будет получен требуемый результат, и это указано на сноске 10 рис. 10.

Доступ к скрытым контактным площадкам теперь можно осуществить через панель **PCB\PCB List**, указав в ней выделение по типу контактных площадок, с признаком **No Net** и (или) обладающими нулевыми размерами. Соответственно здесь же можно восстановить параметры скрытых таким образом контактных площадок, введя их значения, идентичные значениям соседних контактных площадок.

Размещение компонентов

Приступим к размещению компонентов на печатной плате. В связи с конструктивными требованиями положение основных компонентов предопределено и может варьироваться в достаточно узких пределах. Итак, на рис. 12 представлен вариант расположения компонентов. Учтите, местоположение отдельных компонентов определяется в процессе трассировки, и здесь показан конечный вариант расположения:

1. Выходной соединитель. Место определяется привязкой к левому краю печатной пла-

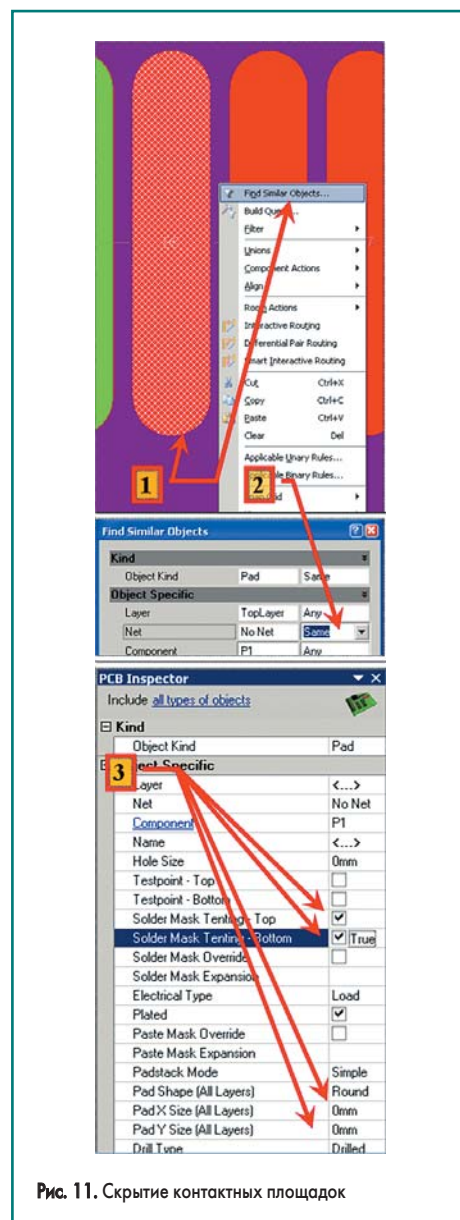


Рис. 11. Скрытие контактных площадок

ты. Варьировать положение можно только по высоте, в пределах, допустимых монтажной планкой печатной платы.

2. Контроллер. Его положение определяется шириной PCI. Возможно изменение местоположения на небольшое расстояние в любом направлении. В нашем примере главное — максимально сдвинуть его вправо и вниз, освобождая наибольшее пространство для топологии трансиверов. Другие компоненты, относящиеся к схеме питания и контроллеру, расположим в правой части печатной платы.

3. Трансиверы. Они расположены с двух сторон от контроллера, максимально разнесены друг от друга, а их порядок определяется расположением выводов контроллера (естественно, так, чтобы минимизировать связи).

4. Кварцевый генератор и другие общие компоненты, которые относятся к трансиверам, разместим между двумя парами трансиверов.

5. Согласующий трансформатор и другие компоненты входных цепей. Трансформаторы расположены симметрично по четыре сверху и снизу печатной платы. Их суммарный размер по длине с учетом зазоров и будет

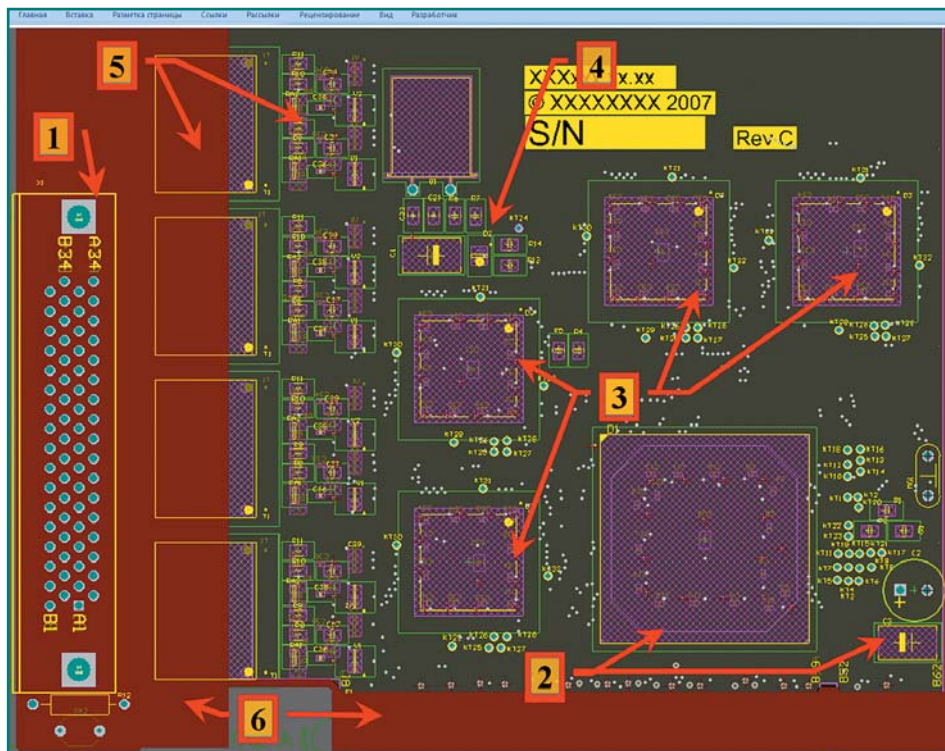


Рис. 12. Размещение компонентов на печатной плате

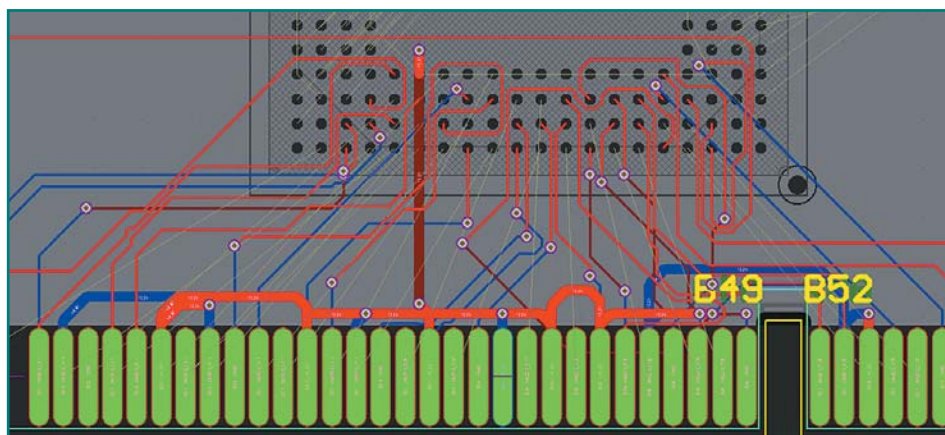


Рис. 13. Автотрассировка компонента

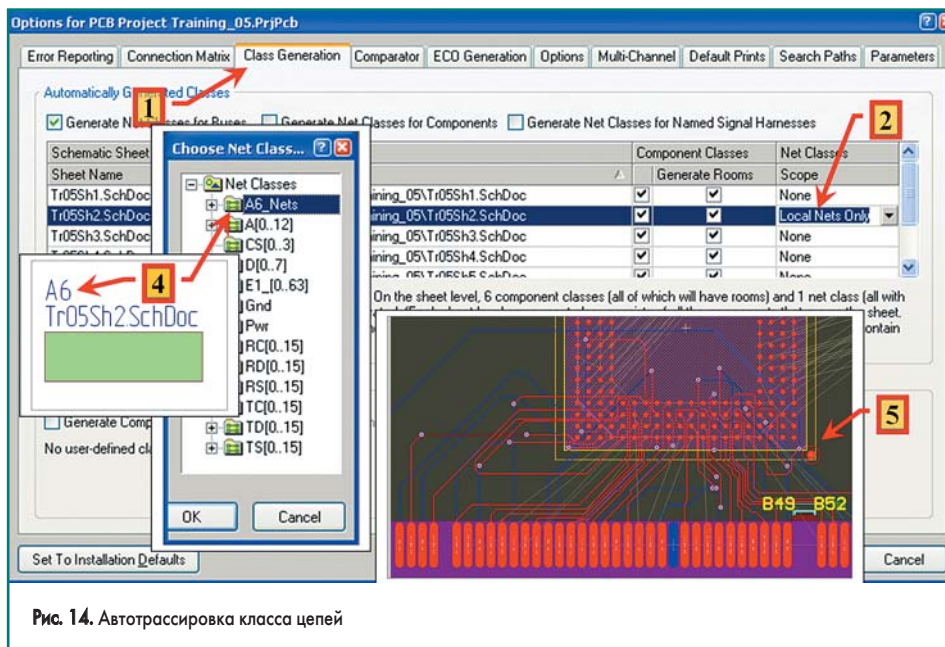


Рис. 14. Автотрассировка класса цепей

определять высоту печатной платы. Из-за того, что одна сторона трансформаторов должна быть обращена к соединителю, их расположение ограничено. С учетом предварительной трассировки их следует максимально приблизить к соединителю, для освобождения большего пространства возле транзисторов. Естественно, порядок трансформаторов определяется порядком расположения транзисторов.

Именно на этапе размещения производится коррекция схемы, а также подключение связей к соединителю — для исключения ненужных пересечений при трассировке.

Топология шины PCI

Остановимся на различных вариантах топологии шины PCI. Для начала выделим компонент D1 (контроллер шины PCI) и поместим его на PCBDOC в зоне печатной платы. Правила для данного проекта не будем приводить. Этот вопрос описан в прежних примерах. Начнем с наиболее простых операций автотрассировки и покажем их действия и возможности.

Автоматическая топология компонента шины PCI

1. Выделим компонент P1 (посадочное место для шины PCI).
2. Нажав правую кнопку мыши, вызываем контекстное меню **Component Action\Autoroute Component** и запускаем автотрассировчик.

На рис. 13 представлен результат работы автотрассировщика. Он не выдерживает никакой критики. Не только не завершена топология электрических связей данного компонента, но и ее качество неудовлетворительно.

Автоматическая топология электрических связей шины PCI

Все электрические связи шины PCI (кроме «земли» и питания) у нас расположены компактно на одном листе схемы, а именно на втором. В этом случае нам легко создать специальный класс для таких связей и затем сделать их автотрассировку. Совершим следующие операции (рис. 14):

1. При открытом листе схемы командой **Project\Project Option** вызываем окно **Option for...** и переходим на вкладку **Class generation**.
2. Для второго листа схемы, содержащей описание электрических связей шины PCI, указываем признак **Local Nets Only** для формирования класса цепей для всех локальных связей на данном листе.
3. Командой **Design Update Pcb...** вносим изменения в файл топологии.
4. Применим команду **Auto Route\Net Class...** для класса цепей **A6_Nets** — именно так назван наш класс (по значению **Designator Sheet Symbol** для этого листа подчиненной схемы).
5. Результат лучше, чем в предыдущем случае, так как отсутствует топология шин питания и «земли», однако и он далек от совершенства.

Не следует сразу убирать созданный класс цепей, так как он в будущем понадобится для написания правил трассировки для данного типа цепей.

Подобным образом можно сделать автоматическую трассировку для:

- выделенной области;
- всех связей между выделенными компонентами;
- всех связей определенной ROOM и т. п.

Однако без существенных внесенных изменений в правила трассировки все эти способы не приведут к результатам, принципиально отличным от приведенных выше. Написание этих правил, а тем более их проверка займут существенно больше времени, нежели ручная или полуавтоматическая трассировка.

Очевидно, таким образом можно производить трассировку только при достаточно большом числе слоев, доступных для трассировки, а для нашего примера они не дадут хорошего результата.

Автоматическое формирование (Fanout) отводов от микросхем

Для симметричных отводов от контактных площадок микросхемы контроллера произведем операцию **Fanout** (рис. 15).

1. Командой **Design\Rules...** вызываем окно **PCB Rules and Constraints Editor**.
2. Открываем стандартное правило: **Design Rule\Routing\Fanout Control\Fanout BGA**.
3. Проверяем или видоизменяем настройки (фрейм **Constraints\Fanout Option** окна) на следующие значения:
 - **Fanout Style** — устанавливаем значение **Auto** (автоматическое определение стиля отвода);
 - **Fanout Direction** — устанавливаем значение **Alternating In and Out** (направление отводов);
 - **Direction From Pad** — устанавливаем значение **Away From Center** (направление отводов относительно контактных площадок);
 - **Via Placement Mode** — устанавливаем значение **Centered Between Pads** (вид расположения переходных отверстий).
4. Выделим компонент D1 (контроллер) и через контекстное меню применим команду **Component Action\Fanout Component**.
5. В результате выполнения операции автоматически будут добавлены переходные отверстия и отводы, как на внешнем слое, так и на внутреннем (отводы именно на этом слое отображены на рисунке). При этом корпус BGA условно разбит на четыре квадранта, и отводы в них сделаны симметрично, от центра.

Получаем ожидаемый результат, который на данном этапе полностью удовлетворяет нашим потребностям, и здесь выигрыш в автоматическом формировании **Fanout** очевиден. После данной операции трассировку производить значительно проще. К сожалению, это выполняется для печатных плат с большим числом слоев.

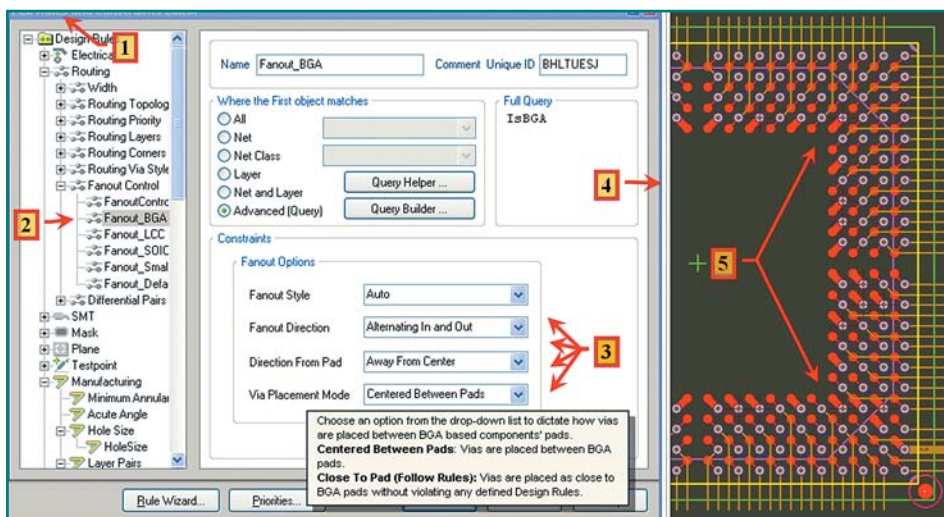


Рис. 15. Формирование Fanout

Полуавтоматическая трассировка шинных электрических связей

1. Выделим две или несколько идентичных электрических связей с отводов от корпуса BGA и применим команду **Place\Multiple Traces** (рис. 16). На рисунке выделено пять электрических связей, с разными правилами для ширины дорожки.
2. Подводим указатель на один из выделенных отводов и делаем трассировку одновременно всех выделенных линий. Поворот всех трасс топологии будет производиться относительно той трассы, к которой изначально был подведен указатель. Трассировка происходит так же, как и для одного сигнала.

Примечание. Для вывода доступных операций при подобной трассировке следует нажать клавишу «~».

Следует обратить внимание: данный способ позволяет быстро прокладывать параллельные трассы. Однако, к сожалению, у него нет функции задания и регулирования зазоров между трассами. В нашем примере зазоры между трассами определены при формировании **Fanout** (отводов от BGA-корпуса) и значительно превышают минимально допустимые.

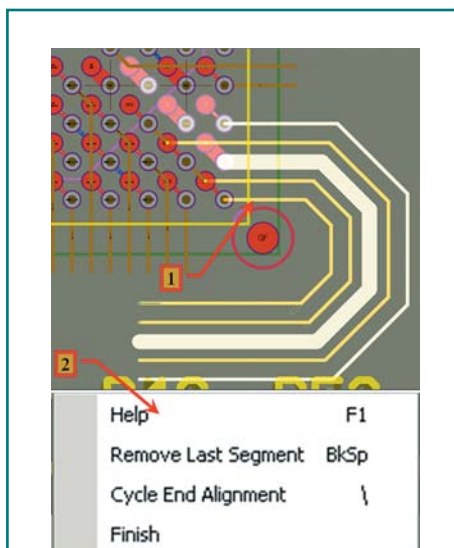


Рис. 16. Трассировка нескольких линий связи

Этот метод рекомендуем применять для трасс, частично подготовленных для такой операции, когда зазоры между трассами уже оптимизированы.

Предложим еще один способ, обладающий более широкими возможностями при одновременной трассировке нескольких связей.

Для этого нам следует удалить все отводы (включая и переходные отверстия) от контактных площадок посадочного места корпуса BGA. Выделим корпус BGA. Нажав правую кнопку мыши, вызовем контекстное меню и применим команду **Component Action\Unroute Component**.

Теперь произведем следующие операции (рис. 17):

1. Выделим несколько контактных площадок и применим команду **Place\Multiple Traces**. Подводим указатель на один из выделенных отводов контактных площадок одновременно всех связей.
2. Перед трассировкой выбор дополнительных операций шире и позволяет устанавливать шаг между трассами.

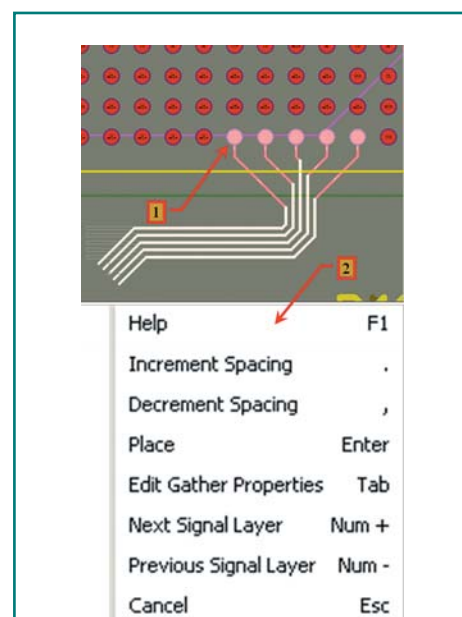


Рис. 17. Трассировка линий от нескольких контактных площадок

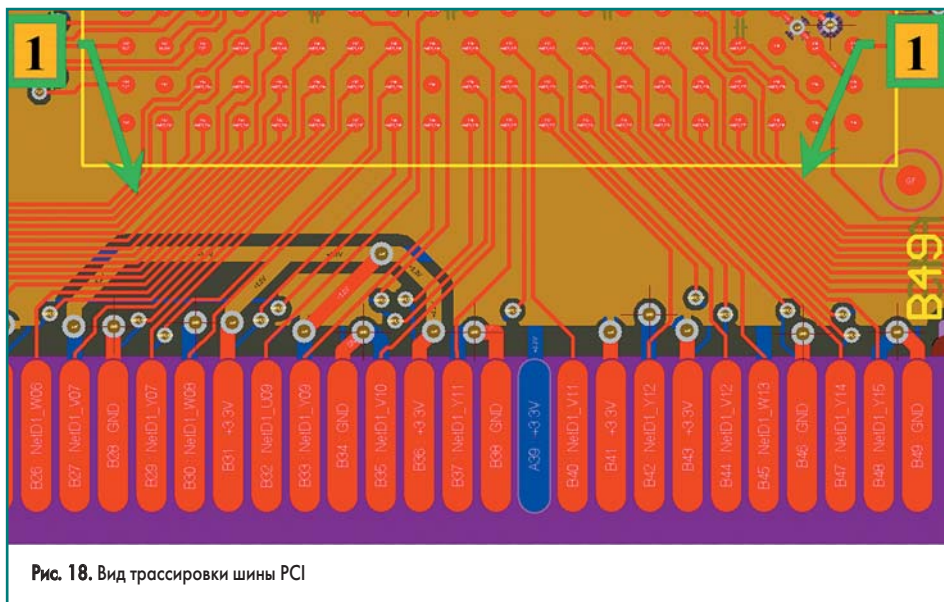


Рис. 18. Вид трассировки шины PCI

При таком применении трассировки шин достаточно просто получить топологию однокорневых связей.

Пропустим операции **Auto Swap** (как в контроллере, так и в шине PCI отсутствуют выводы для такой возможности) и ручной трассировки и представим вид окончательного ре-

зультата (рис. 18). Сноска 1 указывает на зоны параллельной трассировки дорожек при предварительной ручной подготовке.

Другие операции полуавтоматической трассировки (выравнивание трасс, проводка и расчет дифференциальных линий и т. п.) будут рассмотрены в следующих примерах, содер-

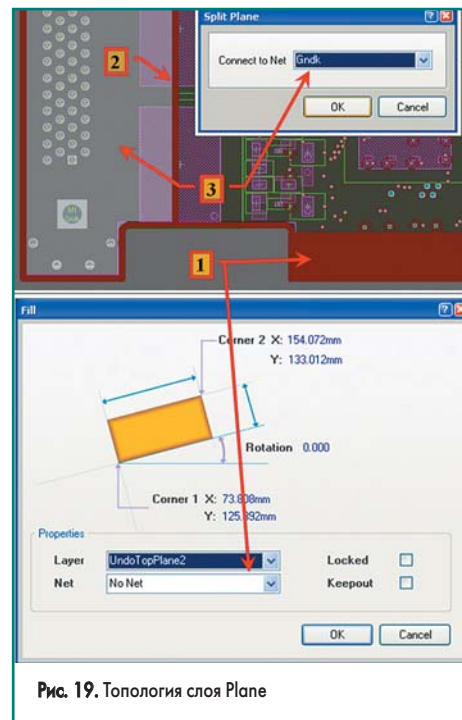


Рис. 19. Топология слоя Plane

жащих схемотехнические блоки, для которых данные операции существенно облегчают трассировку или просто необходимы.

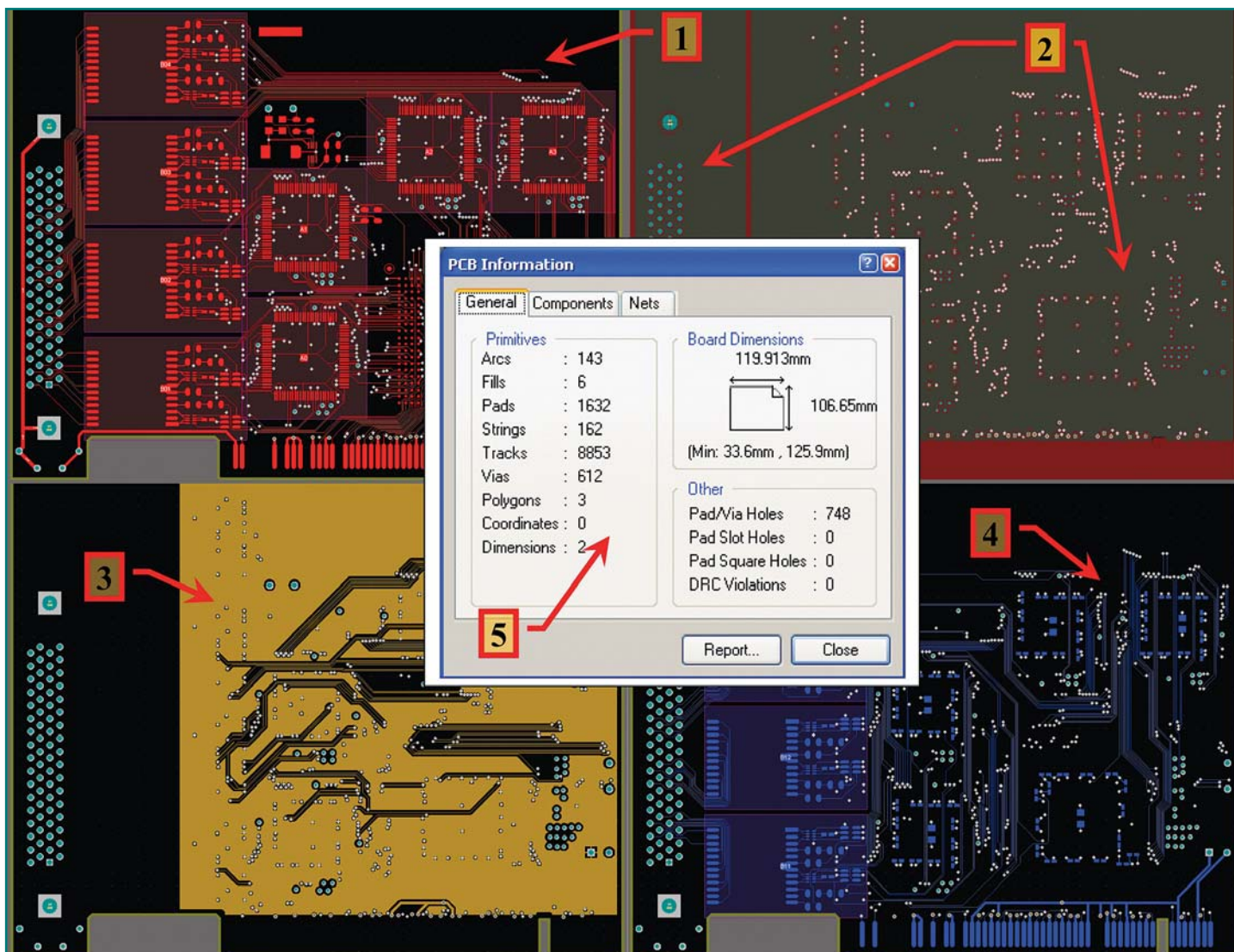


Рис. 20. Послойный вид печатной платы

Общая топология платы

Топология слоя Plane

На слое **Plane** нам следует сделать зоны зашита, во-первых, в области шины PCI, для исключения выхода слоя меди при механической обработке платы (требуется обработка под углом), во-вторых, в области соединителя. В последнем случае можно сделать и по-другому: разделить слой Plane на два участка, и тот, который находится под соединителем, привязать к корпусной «земле».

В первом случае можно обойтись правилами:

- сделать **KeepOut** на этом слое в данной части печатной платы;
- поместить прямоугольный полигон на данном слое. В данном примере мы сделаем именно так.

Для реализации указанного произведем следующие операции (рис. 19):

1. Командой **Place\Fill** разместим в зоне ламелей шины PCI прямоугольник в слое **Plane** и присвоим значение **Net = No Net**. Рекомендуем также установить опцию **Locked**.
2. Командой **Place\Line** разместим от края до края на печатной плате вертикальную линию, которая и разделит слой на две части. Ширину линии следует установить больше допустимых зазоров между двумя новыми цепями. В нашем случае значение ширины равно 1 мм.
3. Выделим часть слоя, которую мы отделили вертикальной линией (на рисунке отмечена более светлым тоном), и присвоим ей новое значение электрической связи — **GndK**.

Топология входных цепей E1

Здесь мы имеем 16 идентичных блоков входных цепей (рис. 6) и выходной соединитель (рис. 4, сноска 4). Методы получения топологии идентичных блоков подробно описаны в [1], и ничего нового мы применять не будем. Отметим только, что и в данном примере:

1. Эта часть печатной платы трассируется с двусторонним расположением элементов.
2. Идентичные блоки сгруппированы по четыре (два блока — на верхней стороне печатной платы, два — на нижней), так как применены 4-канальные трансиверы E1.
3. Топология идентичных модулей получена копированием формата топологии одного из них на остальные.
4. Трассировка топологии связей соединителя сделана в ручном режиме.

Топология трансиверов E1

Электрическая схема состоит из четырех идентичных блоков (рис. 5). Однако, во-первых, трансиверы должны располагаться вокруг (в данном примере — с двух сторон) контроллера шины PCI, во-вторых, они связаны с различными контактными площадками данного контроллера. Это существенно ограничивает преимущества переноса топологии идентичных блоков, но и в данном случае значительная часть работы может быть выполнена именно таким образом. Однако при минимальном количестве слоев основная задача при создании топологии — это ручная трассировка.

Общая топология платы

Напоследок приведем послойную топологию платы для четырех слоев, включая общую информацию о плате (рис. 20).

1. Первый слой (верхняя сторона печатной платы). Размещены основные элементы, плотность топологических элементов — наибольшая.
2. Второй слой. Слой **Plane**. Он предназначен только для общей «земли». Возле соединителя (изолированный сегмент) часть слоя отведена под «землю» корпуса.
3. Третий слой. Виден полигон питания. Используется для трассировки только при невозможности использования внешних слоев.
4. Четвертый слой (нижняя сторона печатной платы). Он предназначен для размещения фильтрующих конденсаторов, а также согласующих трансформаторов и других элементов согласования линий. Вторым по предпочтительности слой для использования в топологии электрических связей.
5. И, наконец, информация, по которой можно судить о плотности топологии.

Предложенный вариант назначения слоев ориентирован на прежние подходы к минимизации слоев и не может быть рекомендован для более сложных, многослойных проектов.

Литература

1. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2007. № 5–8.