

Altium Designer (build 7.x).

Проект многослойной печатной платы: размещение компонентов

В данной статье завершено рассмотрение правил топологии.

Указаны отдельные подходы к способу размещения компонентов на печатной плате.

Владимир Пранович,
к. т. н.

pranovich@bsu.by

Задание правил топологии
в PCB-редакторе

Разделы правил Testpoint и Manufacturing

Опустим рассмотрение правил данных разделов, поскольку они были представлены в предыдущих публикациях, а здесь не претерпели существенных изменений, к тому же эти правила достаточно просты и понятны как в написании, так и по результату действий.

Раздел правил High Speed

Для описываемого проекта многослойной печатной платы было сформулировано несколько правил из данного раздела, и на них мы остановимся подробно. Это связано с тем, что ранее мы их никогда еще не приводили, а в данном примере они решают одну из сложных задач — описание топологии для скоростных сигналов. Мы не будем отражать действия этого раздела в рисунках, поскольку они в основном затрагивают относительно длинные трассы проводников, и журнальные отображения таких трасс будут не отчетливы.

Начнем с подраздела *Parallel Segment*.

- Имя правила — **ParallelSegmentOnSameLayer**. Устанавливает максимальную длину параллельных трасс между любыми проводниками на одном слое.
 - Первое условие — **All**.
 - Второе условие — **All**.
 - Параметры: **Layer Checking = Same Layer, For a parallel gap of = 10 mil, The parallel limit is = 1000 mil**. Здесь указано, что правило действует на проводники, лежащие на одном слое, и что будет произведен контроль максимальной длины (в данном случае не более 1000 mil) параллельных проводников, если расстояние между ними не более 10 mil.
- Имя правила — **ParallelSegment_Allowed**. Для проводников низкоскоростных сигналов устанавливает увеличенное значение максимальной длины параллельных трасс между любыми проводниками, находящимися на одном слое.
 - Первое условие — **InNetClass('LowFreq')**. Указан класс низкоскоростных электрических цепей.
 - Второе условие — **All**.

- Параметры: **Layer Checking = Same Layer, For a parallel gap of = 10 mil, The parallel limit is = 10000 mil**. Установленное значение максимальной длины в 10 раз выше, чем в предыдущем правиле.

- Имя правила — **ParallelSegment_Diff**. Для проводников, принадлежащих разным дифференциальным линиям в одном слое, установлены наиболее жесткие ограничения.
 - Первое условие — **InAnyDifferentialPair**. Указан класс дифференциальных линий.
 - Второе условие — **InAnyDifferentialPair**.

- Параметры: **Layer Checking = Same Layer, For a parallel gap of = 7.5 mil, The parallel limit is = 500 mil**.

- Имя правила — **ParallelSegment_DiffPair_to**. Для проводников, принадлежащих дифференциальным линиям, и проводников других электрических цепей в одном слое установим более жесткие ограничения.
 - Первое условие — **InAnyDifferentialPair**. Указан класс дифференциальных линий.
 - Второе условие — **Not InAnyDifferentialPair**.

- Параметры: **Layer Checking = Same Layer, For a parallel gap of = 20 mil, The parallel limit is = 500 mil**. Здесь установлено значение (20 mil) величины зазора, которое соответствует правилу 4 раздела *Electrical/Clearance* [1], однако оно может быть и больше.

- Имя правила — **ParallelSegment_DiffPair_to Bottom**. Устанавливает зазор между проводниками дифференциальных линий в слое **'2_OverBottom_forDiffPair'** и другими объектами на смежном слое (на соседних слоях — в соответствии с *Layer Stack*, в этом примере это нижний слой **1_Bottom** и слой **3_Ground_Bottom** типа *Plane*, подключенный к электрической цепи *GND*).
 - Первое условие — **InAnyDifferentialPair And OnLayer('2_OverBottom_forDiffPair')**.
 - Второе условие — **OnLayer('1_Bottom') Or InPoly**.

- Параметры: **Layer Checking = Adjacent Layer, For a parallel gap of = 25 mil, The parallel limit is = 500 mil**.

- В подразделе *Length* введем следующие правила, накладывающие различные ограничения, в основ-

Таблица 1. Правила раздела High Speed / Length

Пояснение	Имя правила	Условие	Параметры	
			Min.	Max.
Установим по умолчанию пределы длин проводников. Минимальный размер можно было установить и меньше, здесь главное — максимальный размер, и в данном случае он выбран исходя из габарита печатной платы	Length	All	3 mil	10 000 mil
Установим диапазон допустимых длин проводников данного класса (а именно принадлежащих классу 'PowerLocal', введенному нами ранее [1]) существенно меньшими	Length_PowerLocal	InNetClass ('PowerLocal')	170 mil	3000 mil
Установим пределы длин проводников в этом правиле в еще меньшем диапазоне	Length_PowerLoop	InNetClass ('PowerLoop')	100 mil	1000 mil
Еще один диапазон для другого класса	Length_PowerGlobal	InNetClass ('PowerGlobal');	100 mil	20 000 mil

Таблица 2. Правила раздела High Speed/Matched Net Length

Имя правила	Условие	Погрешность
MatchedLengths_UDVLIO	InNetClass('UD.VLIO')	300 mil
MatchedLengths_BUSVLIO	InNetClass('BUS.VLIO')	300 mil
MatchedLengths_UDDATA	InNetClass('UD.DATA')	250 mil
MatchedLengths_BUSDATA	InNetClass('BUS.DATA')	200 mil
MatchedLengths_UDADR	InNetClass('UD.ADR')	250 mil
MatchedLengths_BUSADR	InNetClass('BUS.ADR')	250 mil
MatchedLengths_XA	InNetClass('XA')	200 mil

ном для проводников питания. Правила сведены в таблицу 1.

В подразделе *Matched Net Length* введем следующие правила (табл. 2), что позволит проверить разброс длин проводников для отдельных классов цепей. При этом во всех правилах флаги **Check Nets Within Differential Pair**, **Check Between Differential Pair**, **Check Between Other Electrical Object** установлены. Заметим, что правила написаны для каждого класса в отдельности, хотя параметры у многих совпадают. Это сделано для удобства корректировки данных правил непосредственно при интерактивной трассировке в сторону более жестких требований, если при создании топологии печатной платы тот или иной класс цепей легко реализовать и для коротких длин проводников.

В подразделе *Daisy Chain Stub Length* введем два правила, определяющие максимальную длину отвода типа «Т» при подключении проводника к контактной площадке.

- Имя правила — **StubLength**. Устанавливает данную длину для всех проводников:
 - Условие — All.
 - Параметры: **Maximum Stub Length** = 1000 mil.
- Имя правила — **StubLength_Diff**. Для проводников дифференциальных линий установим существенно меньшее значение:
 - Условие — **InAnyDifferentialPair**.
 - Параметры: **Maximum Stub Length** = 75 mil.
 В подразделе *Vias Under SMD* введем правила, определяющие возможность совмещения переходных отверстий с контактными площадками планарных компонентов.
- Имя правила — **ViasUnderSMD**. По умолчанию запретим возможность совмещения переходных отверстий и контактных площадок:
 - Условие — All.
 - Флаг **Allow Vias SMD Pads** снят.

- Имя правила — **ViasUnderSMD_PAD_Exception**. Аналогичное правило, созданное для следующего исключения: в проекте для объединения проводников различных электрических цепей используются специальные топологические перемычки, которые не требуют монтажа и которые можно совмещать с переходными отверстиями. В данном проекте имеются контактные площадки большого размера (использованы в посадочных местах силовых компонентов для дополнительного рассеяния мощности), которые требуют наличия переходных отверстий как для снижения плотности тока через переходное отверстие, так и для лучшего отвода тепла через внутренние слои «земли»:
 - Условие —

```
((InNetClass('PowerGlobal')
Or InNetClass('PowerLocal'))
And InPadClass('Heatsink_PAD'))
Or HasFootprint('NET_TIE_10mil'))
```

Здесь указано, для какого класса силовых цепей и класса контактных площадок, а также для какого типа посадочного места будет действовать правило.

- Флаг **Allow Vias SMD Pads** установлен.
- Имя правила — **ViasUnderPAD_QFN**. Правило для еще одного исключения. Посадочное место в корпусе типа QFN, как правило, имеет специальную контактную площадку под корпусом, и ее рекомендовано соединять с теплоотводящими слоями PLANE специальным типом переходных отверстий:
 - Условие — **InPadClass('Heatsink_PAD')**.
 - Флаг **Allow Vias SMD Pads** установлен.
 В подразделе *Maximum Via Count* введем правило, которым ограничим допустимое число переходных отверстий для дифференциальных линий.
 - Условие — **InAnyDifferentialPair**.
 - Флаг: **Maximum Via Count** = 4.

Разделы правил Placement

В данном разделе кроме правил, сформированных по умолчанию в зависимости от установок проекта [1], введем дополнительно еще три в подразделе *Clearance*:

- Имя правила — **ComponentClearance**. Устанавливает зазор между компонентами по умолчанию.
 - Первое условие — All.
 - Второе условие — All.
 - Параметры: флаг **Vertical Clearance Mode** = Specified (установим значения зазора и по горизонтали, и по вертикали); **Minimum Vertical Clearance** = 10 mil, **Minimum Horizontal Clearance** = 3 mil, флаг **Show actual violation distance** установлен.
- Имя правила — **ComponentClearance_CompBd**. Устанавливает зазор для посадочных мест с большими габаритами:
 - Первое условие —


```
HasFootprint('EAS3P127A — V2')
Or HasFootprint('CAPCPD13.5H9.5e5d.9')
Or HasFootprint('SDC013-30-A1')
Or HasFootprint('SOP63P602X173-17L')
Or HasFootprint('B_3B-XH-A')
Or HasFootprint('70ADJ-4MLOG')
Or HasFootprint('SOP63P602X173-17N')
```

 Здесь явно перечислены посадочные места, на которые будет распространяться правило.
 - Второе условие — All.
 - Параметры: флаг **Vertical Clearance Mode** = Specified, **Minimum Vertical Clearance** = 5 mil, **Minimum Horizontal Clearance** = 5 mil, флаг **Show actual violation distance** установлен.
- Имя правила — **ComponentClearance_NoCheck**. Для чисто топологических элементов (не содержащих посадочные места, а только рисунок проводящего слоя) снимем проверку зазора до других посадочных мест. Это позволит размещать такие элементы и под компонентами на печатной плате. Для чего, обратите особое внимание, установим в этом правиле отрицательное значение горизонтального зазора, а проверку по высоте снимем:
 - Первое условие —


```
HasFootprint('LTM0240CA1T0UNDO')
Or HasFootprint('FIDICULAR_40X100')
Or HasFootprint('NET_TIE_10mil')
Or HasFootprint('NET_TIE_5mil')
```
 - Второе условие — All.
 - Параметры: флаг **Vertical Clearance Mode** = Infinite (проверка по вертикали не производится), **Minimum Horizontal Clearance** = -2000 mil (значение отрицательно, а по абсолютному значению больше типовых размеров посадочных мест, под которыми допустимо устанавливать такие элементы), флаг **Show actual violation distance** установлен.
 Оставшиеся разделы правил пока опустим. Итак, мы привели основные правила данного проекта и теперь приступим к размещению компонентов и созданию топологии.

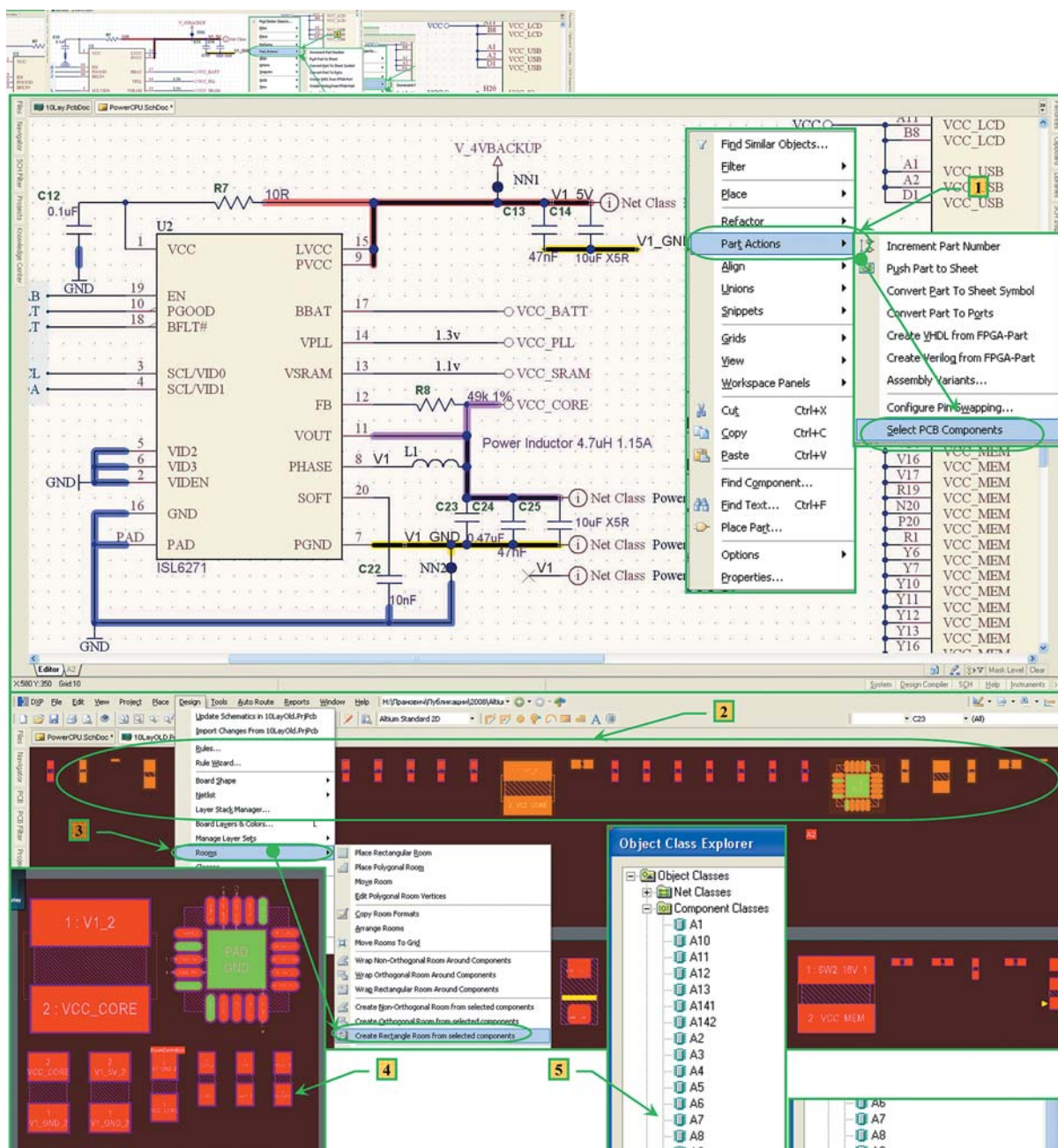


Рис. 1. Схема узла вторичного питания

Расстановка компонентов

Создание Room из компонентов

Размещение компонентов на печатной плате всегда связано с конкретным проектом, и здесь мы укажем только несколько новых приемов, так как эти вопросы частично были рассмотрены в [2, 3].

Итак, обратимся к узлам вторичного питания, для которых мы сформировали (см. рис. 5, 6 в [1]) отдельные классы цепей и компонентов. Рассмотрим узел формирования уровней вторичного питания контроллера, схема которого приведена на рис. 1. Там для данного узла мы вводили только классы электрических цепей. Выделив все компоненты данного узла, можно сформировать и класс компонентов — так, как мы сделали для узлов (см. рис. 6 в [1]), однако здесь приведем еще один способ такой операции:

1. После выделения требуемого узла схемы подводим указатель под один из выделенных компонентов и нажимаем правую кнопку указателя. В контекстном меню применим команду **Part Action >> Select PCB Component**.
2. В результате операции все компоненты узла будут выделены в файле PCB-документа. Это особенно удобно, так как при начальном размещении компоненты сгруппированы только в общей зоне (Room) листа схемы.
3. Не снимая выделения компонентов в файле PCB-документа, применим команду **Design >> Rooms >> Create Orthogonal Room from Components**. При этом будет создана дополнительная прямоугольная область (Room). Переместим данную Room в удобную зону PCB-документа и переопределим ее размеры до необходимой величины. Заметьте: все компоненты нашего узла

при перемещении этой Room будут сдвинуты на такое же расстояние.

4. Для дальнейшей работы удобнее элементы узла питания переместить внутрь созданной Room. Для этого применим команду **Tool >> Component Placement >> Arrange with Room**.
5. Откроем редактор классов (**Design >> Classes...**) и в разделе **Component Classes** найдем только что добавленный класс компонентов “Room Definition 1”. Здесь мы только переименуем указанный класс на определенное нами имя, например **DC/DC_MK**. Напомним: чтобы войти в режим редактирования имени класса, следует нажать левую кнопку указателя.

Таким образом, мы добавили новый класс компонентов и соответствующую ему Room к проекту и определили этот класс только в PCB-документе, но не на схеме. При необходимости аналогично можно создать и другие

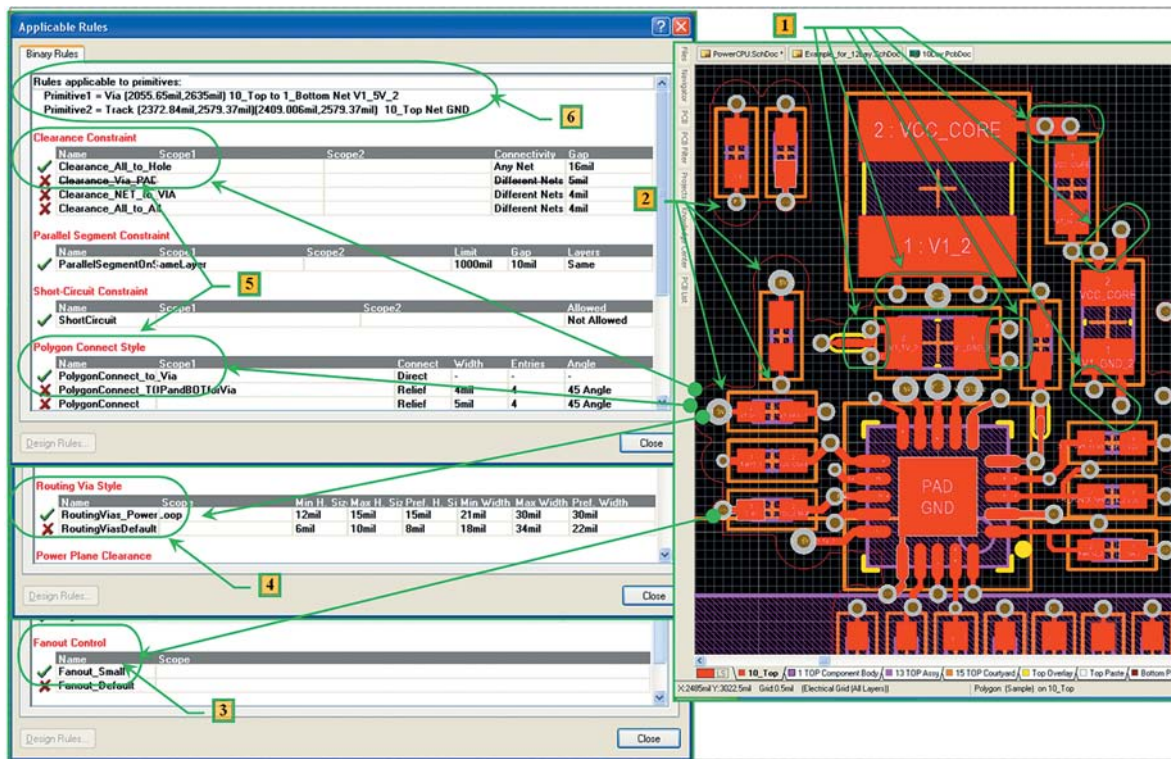


Рис. 2. Размещение и топология узла питания

классы компонентов и скомпоновать их в соответствующей зоне. Однако здесь мы этот процесс опустим.

Теперь следует расположить компоненты нашего узла так, чтобы их положение было оптимальным, с точки зрения топологии. На рис. 2 представлен окончательный вариант расположения этих компонентов, который и будет применен на печатной плате. В частности, на рисунке отмечены:

1. Зоны, где произведено дублирование переходных отверстий и отводов проводников от контактных площадок элементов, через которые протекает ток большой амплитуды. Здесь не использовалась операция **Fanout**.
2. Автоматически сформированные **Fanout** операции проводники и переходные отвер-

стия в соответствии с установленными нами правилами.

3. Для указанного проводника при операции **Fanout** приведено одно из действующих правил. Чтобы определить, какое именно правило будет действовать на проводник (в этом случае — для операции **Fanout**), следует подвести указатель к элементу топологии или компоненту, нажать правую кнопку указателя и в открывшемся контекстном меню выбрать пункт **Applicable Unary Rules**. Аналогично при выборе двух объектов и выборе **Applicable Binary Rules** можно узнать и те правила, которые определяют требования между двумя объектами.
4. Здесь аналогично указано, что при операции **Fanout** размеры переходного отверстия выбраны в соответствии с правилом

RoutingVias_PowerLoop раздела **Routing/Routing Via Style**, который мы задали ранее [4].

5. Здесь указана часть правил (**Binary**), которые определяют стиль подключения переходного отверстия к полигону (он, к сожалению, находится на внутреннем слое и на этом рисунке не отображен) и величину зазора.
6. Для каждого окна **Applicable Rules** приведено описание первого и второго (если он есть) объектов, на которые распространяются те или иные правила. Примечание: правила, которые исключены из проверки, в данном окне зачеркнуты, при этом их порядок в списке определяется приоритетом правила, а действующее правило находится на первой строке своего раздела и имеет флаг зеленого цвета.

Создание Union

Теперь рассмотрим другой пример, где нет необходимости в создании отдельного класса. Например, остановимся на схеме узла последовательного интерфейса RS-232 (рис. 3). Итак:

1. Как и ранее, выделим схему данного узла и применим команду **Part Action >> Select PCB Component**.
2. В данном случае **Room** создавать не будем, а сразу переместим компоненты в требуемую зону и расположим элементы так, чтобы было удобно работать и с топологией, и с дизайном печатной платы.
3. Если мы уверены, что компоненты больше не будут менять взаимного расположения, и для того, чтобы все элементы этого узла перемещались только группой, следует выделить их еще раз и использовать команду контекстного меню (вызов нажатием правой кнопки указателя) **Union >> Create**

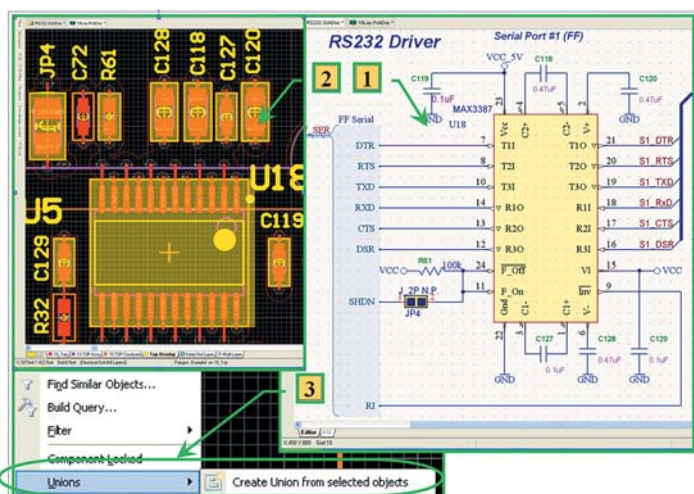


Рис. 3. Пример размещения компонентов последовательного интерфейса

Union from selected object. Если у нас полностью завершена топология данного узла, то в такой **Union** можно и следует включить проводники, полигоны и переходные отверстия, связанные с данным узлом. Теперь все элементы нашего узла в РСВ-документе можно переместить при необходимости в новое место одновременно. При возникновении ситуации, когда все-таки есть необходимость в изменении взаимного расположения элементов топологии в этом узле, аналогичными действиями исключите из **Union** требуемые объекты.

Не станем более приводить примеры расстановки компонентов на печатной плате, так как этот процесс индивидуален для каждого проекта. При необходимости укажем особенности размещения при рассмотрении топологии данного проекта.

Продолжение следует

Литература

1. Пранович В. Altium Designer (build 7.x). Проект многослойной печатной платы: определение классов цепей и компонентов // Тех-

нологии в электронной промышленности. 2008. № 7.

2. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2007. № 6.

3. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2007. № 7.

4. Пранович В. Altium Designer (build 7.x). Проект многослойной печатной платы: правила топологии и результат их действия // Технологии в электронной промышленности. 2008. № 8.