

Altium Designer (build 7.x).

Проект многослойной печатной платы: определение классов цепей и компонентов

В данной статье мы рассмотрим вопросы определения классов цепей и компонентов на электрической схеме. Остановимся только на простых классах, для которых требуется создание правил для топологии или размещения компонентов на печатной плате, а также тех классах, которые упрощают поиск электрических цепей в PCB-редакторе и предоставляют другие удобства в работе.

Владимир Пранович,
к. т. н.

pranovich@bsu.by

Мы не будем касаться всех подробностей работы в редакторе **Schematic Editing**, а остановимся лишь на ключевых моментах, которые принципиальны для данного примера или не были детально рассмотрены ранее.

Структура проекта

В данном примере электрическая схема проекта располагается на нескольких листах и имеет иерархическую структуру. Первый лист (рис. 1) содержит прототип структурной схемы, блоки которой являются ссылками на другие листы схемы проекта. Каждый лист в этом примере, как правило, описывает отдельный узел. Данные узлы имеют свое функциональное назначение, однако не являются самостоятельными и независимыми. В схеме практически нет повторяющихся блоков, и здесь можно присваивать всем именам электрических цепей флаг **Global**. Однако и для этого проекта мы установим все настройки такими, какими они были в прежних примерах (рис. 2),

а здесь еще раз укажем те, что имеют принципиальное значение.

1. Командой **Project >> Project Option** вызовем окно **Option for PCB Project...**, а в нем откроем вкладку **Option**.
2. В зоне **Netlist Option** устанавливаем (снимаем) следующие флаги:
 - Флаги **Allow Port to Name Nets** и **Allow Sheet Enter to Name Nets** сняты. Отсутствие флага запрещает присвоение имен электрическим цепям через имена **Port** или **Sheet Enter**. Все имена цепей в схеме будем задавать через метки **NetName**.
 - Флаг **Append Sheet Numbers to Local Nets** установлен. Это позволит применять одинаковые имена (**NetName**) для различных электрических цепей на разных листах схемы. Такая опция особенно удобна при использовании электрических схем узлов, которые были разработаны для других проектов. При этом схемы могут содержать имена электрических цепей, совпадающие с аналогичными на других листах проекта.

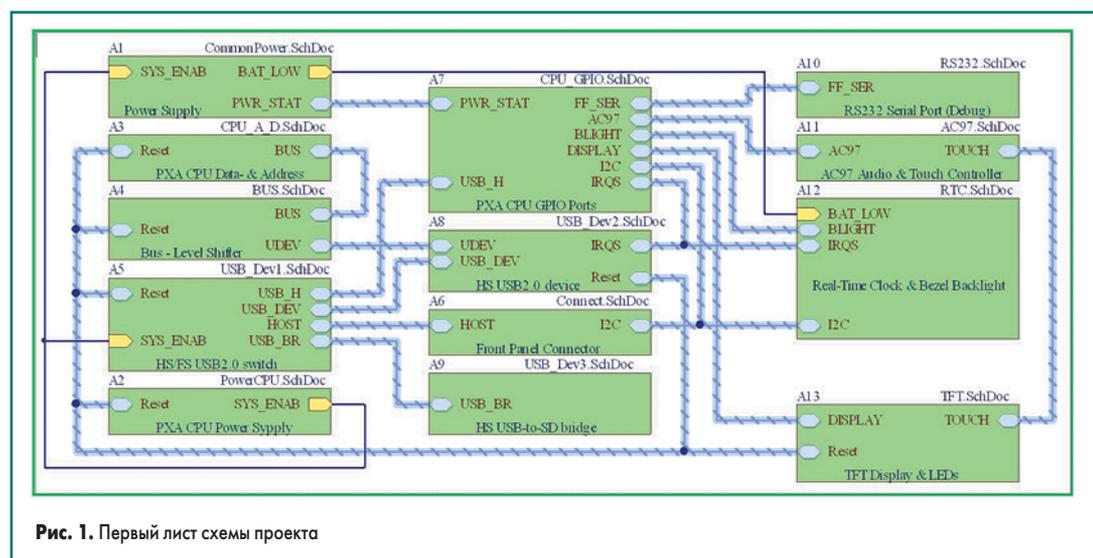


Рис. 1. Первый лист схемы проекта

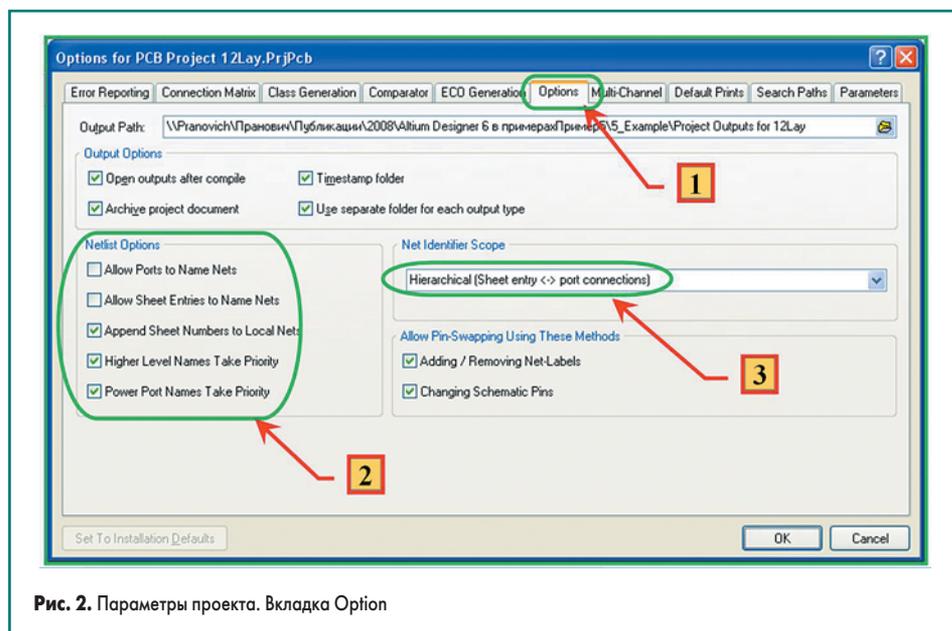


Рис. 2. Параметры проекта. Вкладка Option

Еще раз отметим, что это разные цепи, и они не должны быть электрически объединены в новом проекте. Если данный флаг установлен, к имени электрической цепи будет добавлен номер листа схемы, и таким образом будет произведено разделение цепей, имеющих одинаковые метки **NetName** на разных листах. В этом случае нет необходимости в поиске и переименовании разных электрических цепей с одинаковыми именами на разных листах, если схемы проекта получены путем копирования или переноса листов схем из ранее использовавшихся проектов.

- Флаг **Higher Level Names Take Priority** установлен. Если одной и той же электрической цепи на разных листах схемы присвоено несколько меток **NetLabel** с отличающимися значениями, то в качестве имени цепи будет использовано значение метки **NetLabel** с листа с более высоким приоритетом. Это актуально, так как данный пример проекта содержит много листов со схемами законченных узлов. Еще раз подчеркнем разницу в действии между данным и предыдущим флагом: последний разделяет одну и ту же электрическую цепь проекта, помеченную *одинаковыми* метками **NetName** на *разных* листах схемы; данный устанавливает правило присвоения имени по приоритету для одной и той же электрической цепи проекта, помеченной *разными* метками **NetName** на *разных* листах схемы. Действия установки этого и предыдущего флага рассмотрим более подробно при анализе соответствующих фрагментов схемы.
- Флаг **Power Port Names Take Priority** установлен. Он оказывает аналогичное действие при передаче связей между листами иерархической схемы по пути **Power Port** → **Sheet Enter** → **NetLabel**, когда одной электрической цепи на разных листах схемы присвоено имя с помощью данных меток.

3. Для иерархического проекта устанавливаем соответствующий тип связи: **Net Identifier Scope**; **Hierarchical (Sheet Enter ↔ NetLabel)**.

Задание общих правил проекта

Отметим следующее: в предыдущем разделе статьи [1] мы вводили общие правила для всего проекта, которые учитывают требования производителя печатных плат. Конечно, есть возможность задать эти правила и непосредственно на схеме, при этом они будут распространяться на весь проект. Для этого следует их ввести как параметры проекта со свойством **Rule** (рис. 3).

Итак, для задания общих правил, которые будут распространяться на весь проект (если иное не указано в самом правиле), произведем следующие операции (рис. 3):

1. Откроем первый лист схемы (рис. 1) и командой **Design >> Document Option** вызовем одноименное окно, а затем откроем в нем вкладку **Parameters**.
2. Для добавления правила к проекту следует нажать на кнопку **Add as Rule**.

3. В новом окне **Parameter Properties** для редактирования параметров правила следует нажать на кнопку **Edit Rule Values**. При этом открывается окно **Choose Design Rule Type** (на рисунке не показано), где можно выбрать для редактирования одно из допустимых правил. Список правил и настройка их параметров полностью совпадают с аналогичными из РСВ-редактора. Этот вопрос был подробно освещен в первой части статьи, и здесь отметим только новое. При определении правил через параметры листа остаются недоступными:

- задание имени (**Name**) правила;
- поле комментария (**Comment**) для правила;
- указание приоритета правила.

При передаче данных в РСВ-редактор правила получают имена типа **Schematic Clearance Constraint 1**, где:

- **Schematic** — часть названия, присвоенная по имени редактора, в котором оно введено;
- **Clearance Constraint** — часть названия, присвоенная по имени правила, параметры которого определены в нем;
- **1** — индекс правила, если идентичных правил введено несколько. При добавлении в проект нового правила из данного раздела правил этот индекс для него автоматически будет увеличен на единицу.

Отсутствие возможности самому определять имя правила и его приоритет существенно ограничивает ориентацию в большом числе правил уже в РСВ-редакторе. И именно это обстоятельство, по мнению автора, не дает в полном объеме использовать задание правил для топологии электрических цепей в Schematic-редакторе.

Автоматическая генерация классов

В статье [2] кратко описана последовательность установки опций проекта для автоматической генерации классов. Для данного примера характерно наличие большого разнообразия различных типов классов, поэтому рассмотрим этот вопрос подробнее и предло-

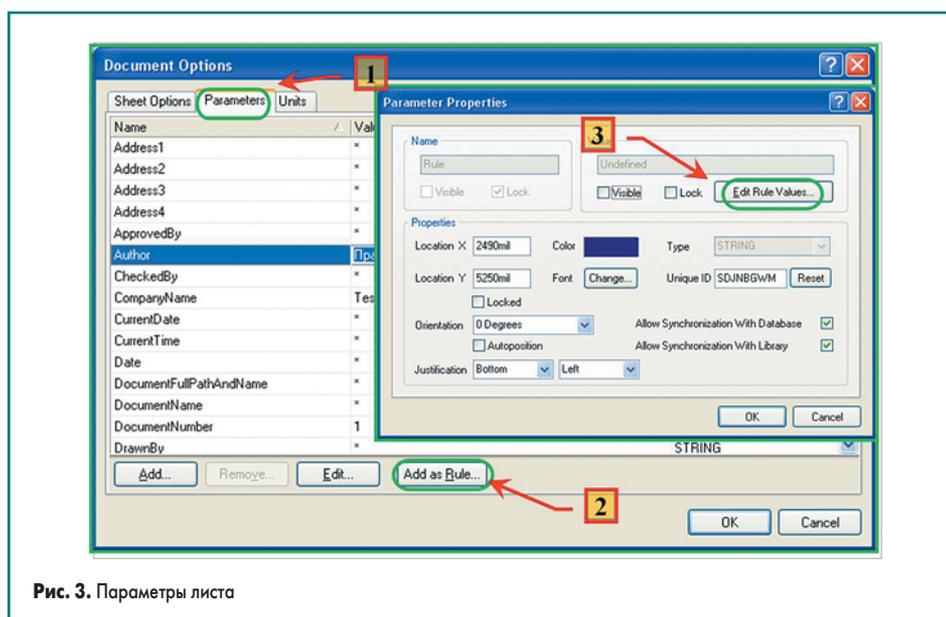


Рис. 3. Параметры листа

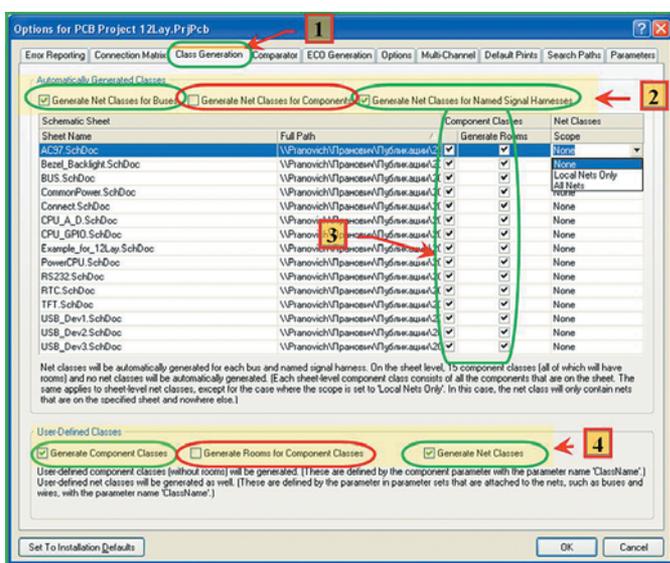


Рис. 4. Параметры проекта. Вкладка Class Generation

жим методы их указания в параметрах проекта, параметрах листов схем или компонентов на ней, а также с помощью специальных директив. Итак, рассмотрим рис. 4:

1. Командой **Project >> Project Option** вызовем окно **Option for PCB Project...**, а в нем откроем вкладку **Class Generation**.
2. В строке **Automatically Generated Classes** установим (снимем) следующие флаги автоматической генерации классов:
 - a) **Generate Net Classes for Busses** — установлен. Для всех **Bus** (шин с одноименными сигналами) будут автоматически сформированы **Net Classes**.
 - b) **Generate Net Classes for Component** — снят. Отдельно для компонентов классы цепей формировать в этом примере не будем.
 - c) **Generate Net Classes for Named Signal Harnesses** — установлен. Для всех шин с разноименными сигналами также укажем автоматическое формирование **Net Classes**.
3. В колонках **Component Classes** и **Generate Rooms** установим все флаги. Флаг в первой колонке означает формирование класса компонентов из указанного листа, во второй — формирование **Rooms**. Использование этих классов и **Rooms** облегчает размещение компонентов на печатной плате, однако здесь мы не будем рассматривать данный вопрос, так как автор раскрыл его в серии предыдущих статей.
4. Аналогично пункту 2 поступим и с установкой флагов формирования классов, которые созданы пользователем:
 - a) **Generate Component Classes** — установлен. Разрешим формирование дополнительных классов компонентов, которые могут быть определены внутри листа схемы.
 - b) **Generate Room for Component** — снят. Не станем формировать несколько **Room** для одного листа. В данном примере все узлы схемы, имеющие самостоятельное применение, и так находятся на отдельных листах.

c) **Generate Net Classes** — установлен. В пределах одного листа может быть сформировано несколько классов цепей.

При таких установках параметров проекта основная масса необходимых классов для работы в PCB-редакторе появится автоматически, необходимо будет добавить только пользовательские классы, а также другие специфические классы, которые сложно задать в SCH-редакторе.

Определение пользовательских классов цепей

Здесь приведем только характерные примеры определения пользовательских классов в SCH-редакторе. Начнем с отнесения электрических цепей к определенным классам. Для этого рассмотрим схему узла DC/DC-преобразователя (рис. 5).

1. Жирными линиями отмечены те электрические цепи, к которым будут предъявлены

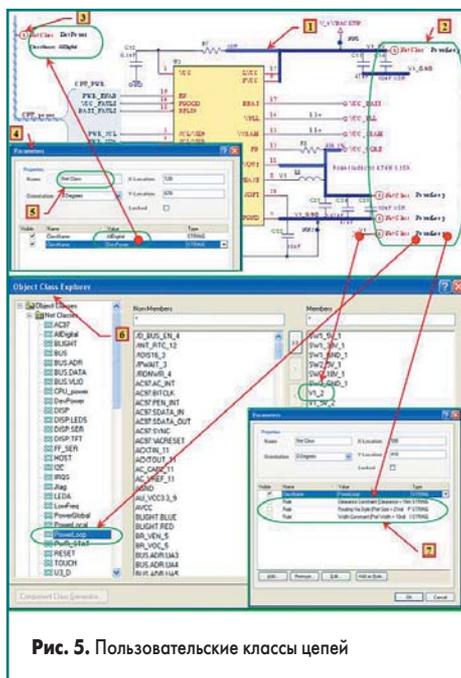


Рис. 5. Пользовательские классы цепей

особые требования при топологии данных узлов DC/DC-преобразователя для снижения импульсных помех. А именно, эти цепи должны обладать:

- достаточной шириной проводников;
- переходными отверстиями с большим значением **Hole** и **Diameter**;
- разрешением дублирования переходных отверстий;
- расположением на специально отведенном внутреннем слое печатной платы.

2. Для отнесения этих цепей к специальному классу на схеме командой **Place >> Directive >> Net Class** введены четыре параметра **ClassName= «PowerLoop»**.

3. Для цепей, входящих в состав шины **Harnees**, достаточно использовать одну директиву, указатель которой и размещен на данной шине.

4. Заметим, что один указатель директивы может содержать несколько параметров типа **ClassName**, которые указывают на принадлежность этой цепи не одному, а нескольким классам цепей. Так на рисунке параметры указателя, размещенного на шине **Harnees**, означают, что все электрические цепи данной шины будут принадлежать двум классам: **AllDigital** и **DevPower**.

5. Указатель директивы может содержать ссылки и на другие типы классов или группы объектов. Для этого достаточно привести описание нового класса или группы. Список стандартных групп можно найти в документе AP0151 Using Design Directives in a Schematic Document. Там же приведены примеры их использования.

Указатель директивы должен быть размещен непосредственно на электрическом соединении или на шинах типа **Bus** или **Harness**. Соответственно действие директивы будет распространяться или на данную электрическую цепь, или на все цепи, входящие в отмеченную шину.

6. Скомпилируем проект и передадим изменения в PCB-редактор. В PCB-редакторе командой **Design >> Classes...** откроем окно **Object Class Employer**. В данном окне специально отмечен созданный на схеме класс цепей. В частности, стрелки на рисунке указывают на:

- созданный класс **PowerLoop**;
- электрическую цепь с локальным именем **V1**, находящуюся на втором листе электрической схемы, которой присвоено глобальное имя **V1_2** в соответствии с установками проекта и которая отнесена к классу **PowerLoop**.

Таким образом, можно полностью определить все классы цепей непосредственно на электрической схеме проекта.

Мы рассмотрели описание лишь для некоторых типов классов, которые пользователь может определить непосредственно на схеме. Все возможные типы классов, какие можно определить на электрической схеме, приведены в таблице 1, где также указаны и иные (не только через директивы) способы задания классов и правил.

Таблица 1. Поддерживаемые в SCH-редакторе определения классов и правил

На каком объекте на схеме установлена директива или заданы параметры	Как и где вводятся параметры	Для каких объектов в PCB формируются правила или классы
Pin (вывод компонента)	Выделить Pin . Нажатием кнопки табуляции войти в окно свойств. Открыть вкладку Parameter . Добавить правило	Pad («ножка» посадочного места)
Port (межлистовой соединитель)	Выделить Port . Нажатием кнопки табуляции войти в окно свойств. Открыть вкладку Parameter . Добавить правило	Net (для электрических связей, к которым подключен Port)
Wire (электрическая связь)	Установить на электрическую связь Wire директиву PCB Layout Directive (Parameter Set object) командой Place >> Directives и в ней добавить правила и класс	Net (для электрических связей, на которую установлена директива)
Bus (шина одноименных электрических связей)	Установить на шину Bus директиву PCB Layout Directive (Parameter Set object) командой Place >> Directives и в ней добавить правила и класс	Net Class (для всех электрических связей, входящих в данную шину)
Harness (жгут разнородных электрических связей)	Установить на жгут Harness директиву PCB Layout Directive (Parameter Set object) командой Place >> Directives и в ней добавить правила и класс	Net Class (для всех электрических связей, входящих в данный жгут)
Component (компонент)	Выделить Component . Нажатием кнопки табуляции войти в окно свойств. В зоне Parameters for... добавить правило	Component (для всех электрических связей, подключенных к компоненту)
Sheet Symbol (указатель на подчиненный лист)	Выделить Sheet Symbol . Нажатием кнопки табуляции войти в окно свойств. Открыть вкладку Parameter . Добавить правило	Component Class (для всех электрических связей и компонентов подчиненного листа)
Sheet (лист схемы)	Командой Design >> Document Options вызвать окно Document Options , а в нем вкладку Parameters . Добавить правило	All Objects (на все объекты проекта)

Определение дополнительных классов компонентов

Классы компонентов автоматически можно создать одновременно с ROOM, как указано на рис. 4, с помощью настроек проекта. Однако они создаются для компонентов, находящихся на отдельных листах электрической, а также для всех листов подчиненных схем иерархического проекта. Иногда возникает необходимость создания разных классов компонентов и для элементов, находящихся на одном листе электрической схемы.

Рассмотрим это на примере листа **CommonPower.SCHDOC**, часть которого представлена на рис. 6.

1. На данном листе находится четыре узла формирования вторичного питания:

- a) DC/DC-стабилизатор понижающего типа;
- b) идентичный DC/DC-стабилизатор понижающего типа на другой уровень питания;
- c) узел линейных стабилизаторов для работы от аккумуляторных батарей;
- d) линейный стабилизатор.

Очевидно, что компоненты данных узлов можно отнести к разным классам компонентов, так как есть различия в топологических подходах к реализации этих узлов. Соответственно могут потребоваться специфические правила для описания требований, учитывающих эти различия.

2. Сделаем из компонентов узла 1a отдельный класс. Для этого к компонентам узла добавим новый параметр и сделаем это, например, так:

- выделим все компоненты этого узла;
 - вызовем инспектор (горячая клавиша **F11**);
 - нажмем ссылку **all types of object**;
 - в открывшемся окне установим флаг **Display only**, а затем в ставшем доступным окне флаг **Part**;
 - возвращаемся в меню инспектора **SCH Inspector** и в ставшей доступной строку **Add User Parameter** раздела **Parameters** вносим новое название класса для выделенной группы компонентов, например, по названию формируемого вторичного напряжения питания: **Power_VCC_5V**;
 - после ввода значения параметра программа предложит дать имя новому параметру. Сюда мы вносим название **ClassName**.
3. Теперь при экспорте изменений из схемы в PCB наш класс компонентов, созданный на схеме, будет внесен в соответствующий раздел классов PCB-проекта.
4. В списке классов компонентов отмечен только что созданный класс **Power_VCC_5V**, а также указан список из всех элементов, которые вошли в данный класс.

Задание на схеме правил топологии

В редакторе непосредственно на электрической схеме можно не только определять классы цепей и компонентов, но и сразу задавать правила для топологического редактора. Рассмотрим это на классе, который мы определили ранее. На сноске 2 (рис. 5) указана директива, которая относит локальную цепь **V1** к классу цепей **PowerLoop**. Откроем свойства этой директивы и добавим сразу и параметры правил для топологического редактора. Указанный класс цепей определен для цепей высокочастотных DC/DC-преобразователей с высокой плотностью тока. Приведенные далее правила ориентированы на минимизацию излучений от проводников электрических цепей данного класса.

Все правила при отображении на схеме записываются каждое в своей строке. Так как длина строки при сложных правилах может иметь большой размер, да и самих правил может быть много, нет смысла отображать их на схеме, и следует снять флаг их видимости (рис. 5, сноска 4, окно **Parameters**, флаг **Visible**). Однако все же следует дать разное отображение текстовой записи на директиве, чтобы различать последние с различным наполнением. Поэтому укажем имя для нашей директивы, например, такое: **01ClassNameAndRule**, или еще более кратко, в виде аббревиатуры: **01CNR**. После этого и на другие цепи, на которые должны распространяться данные правила, можно установить эту директиву простым копированием.

Итак, для указанной директивы сформируем следующие правила и в том порядке, в котором они доступны в окне **Choose Design Rule Type** (рис. 7). Данное окно открывается через вызов свойств директивы нажатием кнопки **Add as Rule** и затем в окне **Parameter Properties** кнопки **Edit Rule Values**. Правила для данной директивы будут таковы:

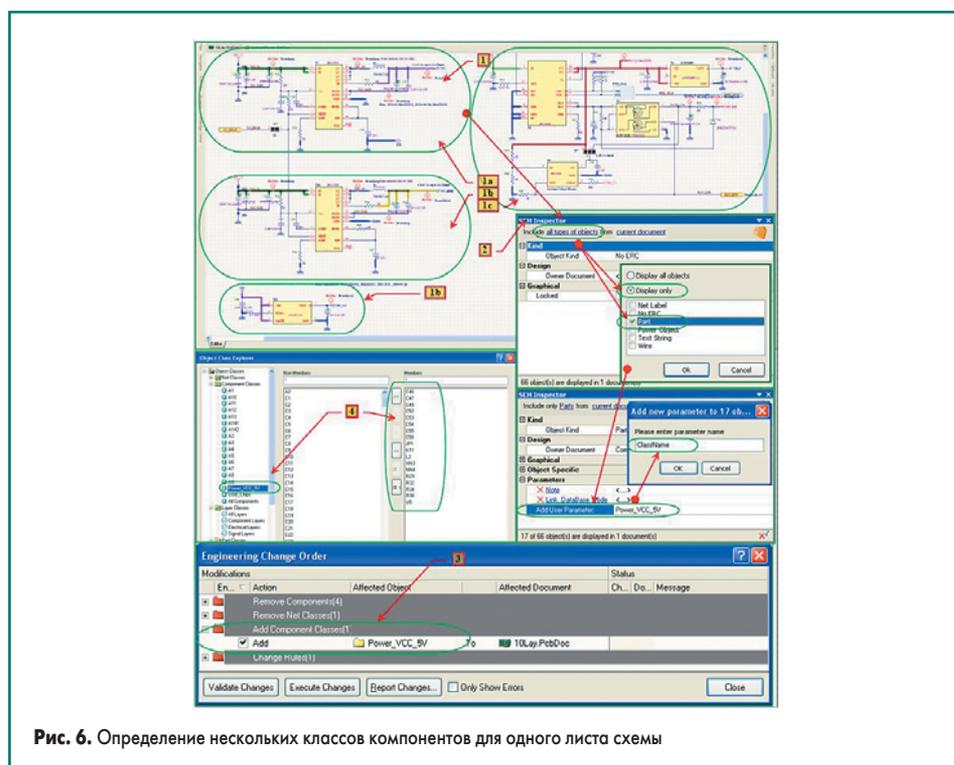


Рис. 6. Определение нескольких классов компонентов для одного листа схемы

а) Установим увеличенный зазор (0,2 мм) между элементами топологии, принадлежащими данному классу, и другими проводниками на печатной плате:

- раздел правил **Electrical/Clearance Constraint**;
- параметры:

Any Net; **Minimum Clearance** = 8 mil.

б) Зададим минимальную ширину проводников (а также и максимальную, и используемую по умолчанию). В нашем случае она будет увеличена, так как директива определяет топологию проводников с высокой плотностью тока:

- раздел правил **Routing/Width Constraint**;
- параметры: **Min Width** = 8 mil;
- Preferred Width** = 10 mil;
- Max Width** = 16 mil.

Примечание. Параметры установим для всех слоев одинаковыми.

с) Определим тип переходных отверстий:

- раздел правил **Routing/Routing Via Style**;
- параметры:
- Via Diameter** — **Minimum** = 21 mil;
- Maximum** = 30 mil; **Preferred** = 21 mil.
- Via Hole** — **Minimum** = 12 mil;
- Maximum** = 15 mil; **Preferred** = 12 mil.

Мы ограничимся этими правилами и укажем их вид уже непосредственно в PCB-документе. Командой **Design >> Update PCB Document...** внесем новые правила в файл топологии. Командой **Design >> Rules...** при активном PCB-документе откроем окно **PCB Rules and Constraints Editor** (рис. 7.):

1. Правило зазора **Clearance Constraint**, введенное в пункте а):

- находится в разделе правил **Electrical/Clearance Constraint**;
- имеет имя **Schematic Clearance Constraint**. *Примечание.* Каждое новое правило подобного типа получит имя **Schematic Clearance Constraint №№**, где № — порядковый номер по мере возрастания;
- правило получает наивысший приоритет как введенное последним;
- поле **Comment** остается незаполненным;
- имя правила и значение поля **Comment** можно изменить, однако значения данных параметров обратно в схему не передаются;
- в первом условии находится выражение **InNet('V1_2')**, означающее, что правило действует на ту электрическую цепь, на которую была установлена директива на схеме. Цепь, помеченная как 'V1' на схеме (рис. 5), в PCB-документе получит имя 'V1_2', в соответствии с настройками проекта (рис. 2, сноска 2);
- остальные параметры правила имеют значения, которые были установлены на схеме с помощью директивы.

2. Правило задания типа переходных отверстий **Routing Via Style**, введенное в пункте с):

- находится в разделе правил **Routing/Routing Via Style**;
- имеет имя **Schematic Routing Via Style**;
- приоритет, поле **Comment** и первое условие правила — аналогичны предыдущему пункту;

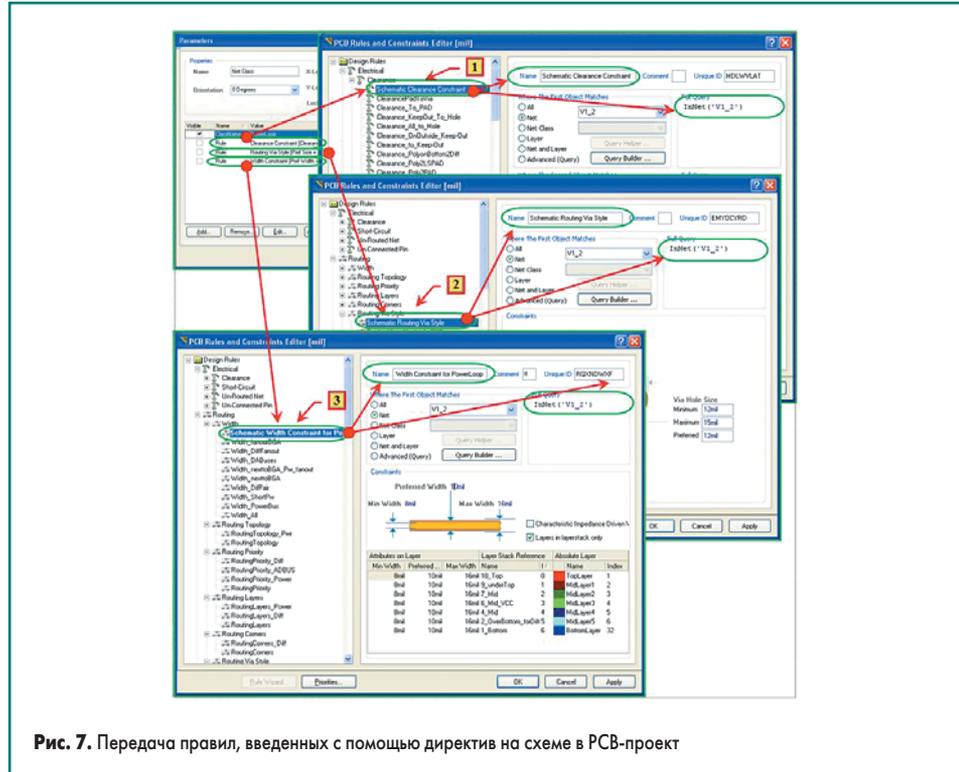


Рис. 7. Передача правил, введенных с помощью директив на схеме в PCB-проект

– остальные параметры правила имеют те значения, которые были установлены на схеме с помощью директивы.

3. Правило задания ширины проводников **Width Constraint**, введенное в пункте б):

- находится в разделе правил **Routing/Width**;
- имеет имя **Schematic Width Constraint for PowerLoops**;
- приоритет, поле **Comment** и первое условие правила — аналогичны предыдущим пунктам;
- остальные параметры правила имеют те значения, которые были установлены на схеме с помощью директивы.

Для анализа правил, введенных через директиву, этого достаточно, так как иные правила будут иметь те же особенности, что отмечены ранее.

Итак, анализируя правила, введенные на схеме с помощью директив, отметим следующее:

1. Имена правил, введенных через директивы, имеют автоматически сформулированное название, что затрудняет ориентацию по этому признаку.
2. Отсутствие возможности ввода комментария также не способствует пониманию сути и назначения правила.
3. Присвоение последнему введенному правилу более высокого приоритета очевидно, однако изменить его непосредственно в директиве нельзя. Эту операцию можно легко установить уже только в PCB-редакторе. Это менее удобно при задании правил через директивы, а также затрудняет оценку на схеме, какое из правил более приоритетно.
4. Существенным недостатком является то, что правило, введенное через директиву, действует только на одну электрическую цепь, но не на весь класс в целом. Расположив директиву на **Harness**, вы будете иметь в PCB-документе столько правил с идентичными параметрами, сколько электрических свя-

зей входит в жгут **Harness**. Таким образом, их общее количество в PCB-документе становится невероятно большим.

5. Существенным недостатком является и то, что нельзя непосредственно в правилах указать иные цепи или классы цепей с идентичными параметрами, на которые директива не установлена.

Этого вполне достаточно, чтобы отказаться от широкого использования директив на электрических схемах. Их следует, по мнению автора, использовать только для наиболее принципиальных цепей. Далее более подробно остановимся только на применении специализированных директив для дифференциальных линий.

Учет класса точности

В первой части статьи мы приводили правила, которые учитывают требования производителя печатных плат. Сейчас обратимся к аналогичному вопросу и учтем требования класса точности печатной платы. Заметим, это разные требования, и если вы разрабатываете топологию под определенный класс точности печатной платы, вы должны выбрать производителя, технология которого может обеспечить данный уровень сложности.

Основные требования к печатным платам приведены в ГОСТ 23751 «Платы печатные. Основные параметры конструкции». Буквенные обозначения размеров основных конструкций печатных плат приведены на рис. 8, где:

- H_{nc} — суммарная толщина печатной платы;
- H_n — толщина печатной платы;
- H_m — толщина материала основания печатной платы;
- h_{ϕ} — толщина фольги;
- b — гарантийный «поясок» сквозного отверстия;

- t — ширина печатного проводника;
- S — расстояние между краями соседних элементов проводящего рисунка;
- Q — расстояние от края печатной платы, выреза, паза до элементов проводящего рисунка;
- d — диаметр отверстия;
- D — диаметр контактной площадки;
- h_n — толщина химико-гальванического покрытия;
- h — толщина проводящего рисунка;
- l — расстояние между центрами (осями) элементов конструкции печатной платы.

Практически все эти параметры мы учитывали, когда описывали правила учета технологических требований производителя печатных плат. Теперь обратимся к подобным с точки зрения требований ГОСТа. В данном примере мы будем разрабатывать топологию по требованиям пятого класса.

В таблице 2 указаны наименьшие номинальные значения основных размеров элементов конструкции печатных плат для узкого места в зависимости от классов точности. Для участков печатной платы в свободных местах указанные значения можно устанавливать по любому более низкому классу, а для первого класса — увеличивать в два раза.

Итак, добавим новые правила, которые будут действовать на все объекты топологии, вне зависимости от классов цепей, и они будут действовать как **Default** (по умолчанию). Приоритет данных правил будет самым низким, однако выше, чем правила учета требований производителя.

Правила приведем в той последовательности, как перечислены значения параметров конструкции в таблице 2.

Таблица 2. Основные размеры элементов конструкции печатных плат

Условное обозначение	Номинальное значение основных размеров для класса точности				
	1	2	3	4	5
t , мм	0,75	0,45	0,25	0,15	0,10
S , мм	0,75	0,45	0,25	0,15	0,10
b , мм	0,30	0,20	0,10	0,05	0,025
γ	0,40	0,40	0,33	0,25	0,20

Примечание. γ — отношение номинального значения диаметра наименьшего из металлизированных отверстий к толщине печатной платы.

1. Правило ширины печатного проводника для пятого класса точности. Раздел правил **Routing/Width:**

- имя (Name) правила — **Width_All_Class5;**
- первое (The First Object) условие — **ALL;**
- параметры: **Min Width = 4 mil;**
Preferred Width = 6 mil;
Max Width = 8 mil.

Это правило полностью совпадает с аналогичным правилом **Width_All**, приведенным в первой части статьи, так как производитель печатных плат и обеспечивает изготовление их по пятому классу. Далее совпадающие или идентичные правила мы приводить не будем.

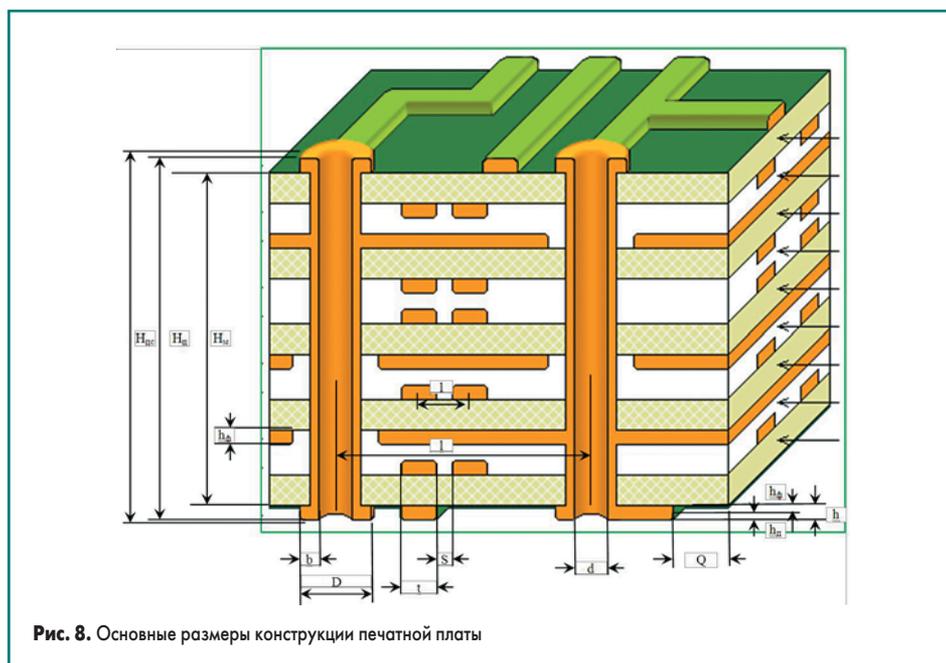


Рис. 8. Основные размеры конструкции печатной платы

Идентично правило ширины печатного проводника для четвертого класса точности:

- имя правила — **Width_All_Class5;**
- первое условие — **ALL;**
- параметры: **Min Width = 4 mil;**
Preferred Width = 6 mil;
Max Width = 8 mil.

2. Правило зазора (расстояние между краями соседних элементов проводящего рисунка) для пятого класса точности:

- раздел правил **Electrical/Clearance;**
- имя правила — **Clearance_AllToAll_5Classes;**
- первое условие — **ALL;**
- второе условие — **ALL;**
- параметры: **Different Net Only;**
Minimum Clearance = 4 mil.

3. Задание гарантированного «пояска» металлизированного отверстия. Отметим сразу, однако, что выбранный нами производитель печатных плат обеспечивает их надежность при рамере гарантированного «пояска» не менее 0,1 мм, что соответствует всего лишь 3 классу точности согласно таблице 2. В этом смысле нет необходимости задавать правила для более высоких классов и вполне можно оставить введенные ранее правила. Однако обратимся к таблице 3. В ней приведены отклонения диаметров сквозных отверстий в зависимости от класса. Так как сверление и наложение фотшаблонов — это разные операции, чтобы избежать дефекта неполной металлизации сквозных отверстий, следует дополни-

Таблица 3. Предельные отклонения диаметров монтажных и переходных отверстий

Наличие металлизации	Предельное отклонение диаметра Δd , мм, для класса точности				
	1	2	3	4	5
Без металлизации	$\pm 0,10$	$\pm 0,10$	$\pm 0,05$	$\pm 0,05$	$\pm 0,05$ ($\pm 0,025$)
С металлизацией без оплавления	+0,05; -0,15	+0,05; -0,15	+0; -0,10	+0; -0,10	+0;-0,10 (+0;-0,075)
С металлизацией и с оплавлением	-	+0,05; -0,18	+0; -0,13	+0; -0,13	+0;-0,13

тельно прибавить к величине гарантированного «пояска», приведенного в таблице 2, и величину отклонения диаметров сквозных отверстий. Теперь, как и ранее, введем два правила, а именно для переходных отверстий и для штыревых элементов. Итак, в разделе правил **Manufacturing/Minimum Annular Ring:**

a) первое правило — **MinimumAnnularRing-ForVia_Class5** (0,025+0,1 мм);

- первое условие — **IsVia,**
- для всех переходных отверстий;
- параметры:

Minimum Annular Ring (x-y) = 5 mil.

b) второе правило — **MinimumAnnularRing-ForPAB_Class4** (0,05+0,1 мм);

- первое условие — **IsPad,**
- для всех переходных отверстий;
- параметры:

Minimum Annular Ring (x-y) = 6 mil.

4. Правило отношения номинального значения диаметра наименьшего из металлизированных отверстий к толщине печатной платы:

- раздел правил **Manufacturing/HoleSize;**
- первое правило — **HoleSize_Class5_H1.6mm;**
- первое условие — **All;**
- параметры: **Measurement Method = Percent** (задание отношения в процентах); **Minimum = 20%** (величина из таблицы 2); **Maximum = 80%** (величина, не имеющая отношения к классу, однако ограничивающая и максимальный размер).

Продолжение следует

Литература

1. Пранович В. Altium Designer (build 7.x). Проект многослойной печатной платы // Технологии в электронной промышленности. 2008. № 6.
2. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2008. № 1.