

Altium Designer Summer 08 – Разработка правил проектирования для печатной платы

(продолжение)

Алексей Сабунин (Москва)

В продолжении статьи описывается процедура создания правила и разобраны примеры создания сложных правил с помощью запросов (Query).

Порядок создания правил

Прежде чем открыть окно Design Rules и приступить к описанию правил проектирования, убедитесь,

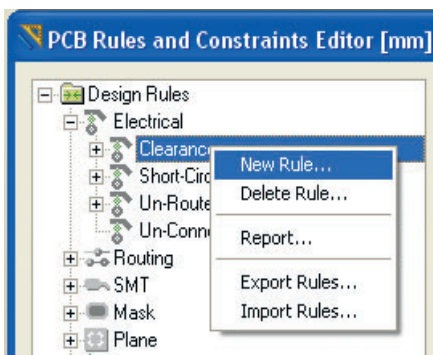


Рис. 2. Создание нового правила

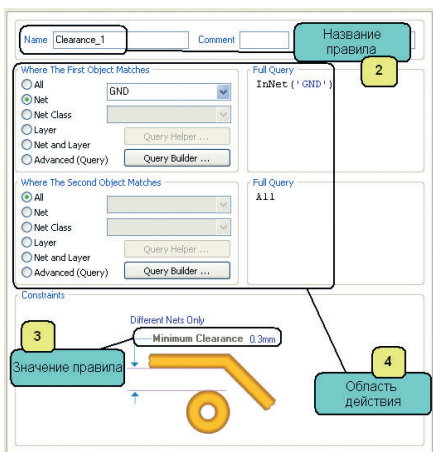


Рис. 3. Описание нового правила (2, 3 и 4 шаги)

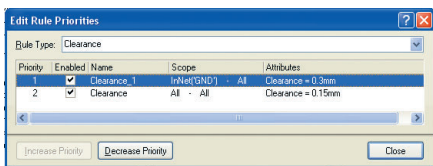


Рис. 4. Приоритеты выполнения правил

что работа ведётся в метрической системе координат, в противном случае значения правил будут отображаться в милсах (1 mil = 0,254 мм). Для включения метрической системы координат используется строка *Units* диалогового окна *Design > Board Options*. Далее опишем порядок создания нового правила по шагам.

Шаг 1 – создадим новое правило. Для этого в соответствующей категории нажимаем правую кнопку мыши и выбираем команду *New Rule* (см. рис. 2). В списке правил появилось новое правило, которое по умолчанию названо *Clearance_1*.

Шаг 2 – присвоим правилу уникальное название. Перейдём к редактированию правила, выбрав его из списка. Название правила вводится в поле *Name* (см. рис. 3), причём все правила должны иметь уникальные названия, иначе программа не сохранит созданные правила.

Шаг 3 – указание области действия правила. Для выбранного нами правила (*Clearance*) необходимо указать, между какими цепями будет действовать указанное правило. Область действия может быть указана вручную или с помощью языка запросов *Query Helper* и составителя запросов *Query Builder*. При ручном выборе области действия правила могут быть выбраны следующие варианты:

- All – все проводники платы;
- Net – одна выбранная цепь;
- Net Class – класс цепей;
- Layer – все цепи, расположенные на данном слое;
- Net and layer – определённая цепь на данном слое.

В нашем случае зададим правило, указывающее зазор между цепью GND, которая будет выполнена полигоном на внешних и одном из внутренних слоёв платы. Название правила пока остаётся без изменения, а в области действия следует указать цепь GND. Случаи, когда область действия задаётся с помощью языка запросов (*Query*), будут рассмотрены ниже.

Шаг 4 – установка значения правила. Для этого используется нижняя часть окна *PCB Rules* (см. рис. 3), причём для большинства правил имеется иллюстрация, поясняющая сущность правила. В нашем случае необходимо написать значение зазора между цепью GND и остальными проводниками платы.

Шаг 5 – установка приоритетов правил. Области действия различных правил могут пересекаться между собой. Например, цепь земли, которую мы описали ранее, входит в наше правило, но также она участвует в базовом правиле, которое задаёт зазоры на все электрические примитивы на плате. Какое правило будет использовать программа *Altium Designer*?

При создании правил в программе P-CAD приоритеты были заданы по умолчанию и правила выполнялись от меньшего объекта к большему, т.е. сначала выполнялось правило на цепь (*Net*), затем на класс цепей (*Net Class*) и в последнюю очередь на всю плату.

В программе *Altium Designer* для установки приоритетов используется специальное диалоговое окно *Edit Rule Priorities* (см. рис. 4), которое вызывается кнопкой *Priorities* в левом нижнем углу окна *PCB Rules*. Правила выполняются в том порядке, в котором они перечислены в списке окна *Edit Rule Priorities*, причём если цепь GND уже указана в пер-

вом по порядку правиле, то она автоматически исключается из всех последующих правил данной группы. Заканчивается процедура создания правил нажатием кнопки Apply в окне создания правил.

Теперь рассмотрим процесс формирования более сложных правил. Когда необходимо в области действия правила указать несколько однотипных объектов, не связанных между собой, можно использовать составитель запросов Query Builder, для чего используется одноименная кнопка (см. рис. 3).

При выборе Query Builder появляется окно Builder Query From Board (см. рис. 5), в котором работа ведётся следующим образом: в списке Condition Type/Operation выбираем тип объекта Belong to Net, а в списке Condition Value – значение параметра (в данном случае, название цепи) CFG_DIN. Аналогично добавляем вторую цепь CFG_INIT. В результате в область действия будут попадать две выбранные цепи. Конструктор запросов предлагает выбрать из списка критерии, по которым будут выбираться объекты применения данного правила. Здесь можно указать определённые цепи или классы на заданных слоях, принадлежность к компонентам или посадочным местам и т.д.

Более тонкая настройка правил проектирования выполняется с помощью окна Query Helper. Давайте рассмотрим использование этой возможности на конкретных примерах.

Примеры создания сложных правил с помощью языка запросов Query Helper

Пример 1. Шаг выводов микросхемы U1 – 0,2 мм, а минимальный зазор между объектами печатного монтажа 0,25 мм, т.е. программа выдаёт ошибку по зазорам между выводами этого компонента и не позволяет прокладывать проводники от выводов данной микросхемы. Задача: указать в правилах, что для контактных площадок компонента U1 разрешён зазор 1,7 мм.

Для решения этой задачи создадим новое правило в группе Clearance. Укажем название данного правила Clearance_FPGA_Pins. Рекомендуется оставлять заголовок правила (в нашем случае Clearance_) и далее в названии кратко описать область его

действия. В этом случае правило становится понятным любому разработчику.

Далее, для указания области действия правила выбираем параметр Advanced (Query), после чего становится активной кнопка Query Helper, которую и следует выбрать. В появившемся окне (см. рис. 6) указаны критерии запроса и операции, которые к ним можно применить. Например, выберем группу Membership Checks и в ней команду InComponent. Эта команда позволяет указать принадлежность каких-либо объектов указанному компоненту. Чтобы узнать синтаксис использования данной команды, следует после выбора команды в списке нажать клавишу F1, которая показывает контекстную справку; в данном случае будет отображён синтаксис описания данной команды:

Examples

```
InComponent('U1')
InComponent('U1') = True
Returns any component that has a Name property of 'U1', along with all of the child objects of each such component.
InComponent('U1','U2','J1')
InComponent('U1','U2','J1') = True
Returns any component that has a Name property of 'U1' or 'U2' or 'J1', along with all of the child objects of each such component.
InComponent('U*')
InComponent('U*') = True
Returns any component that has a Name property whose associated string starts with 'U', along with all of the child objects of each such component.
```

Справка содержит несколько примеров описания данной команды: например, как задать один компонент, три компонента и все компоненты, относящиеся к одному типу. Нас интересует первый вариант. Полный запрос должен выглядеть, как показано на рисунке 6, и по его результатам будут определены все контактные площадки (IsPad), относящиеся к компоненту U1. После составления запроса его необходимо проверить нажатием кнопки Check Syntax. Следует отметить, что проверяется только форма запроса, а не логическая правильность.

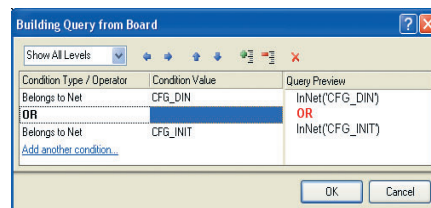


Рис. 5. Конструктор запросов (Query Builder)

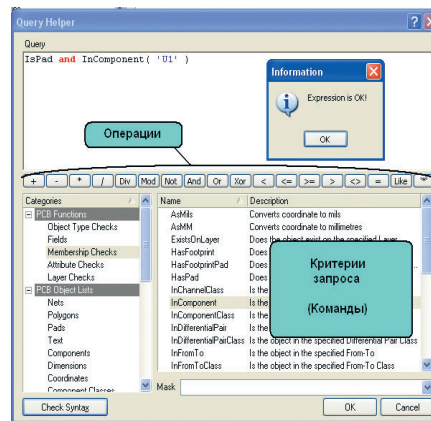


Рис. 6. Создание сложного запроса

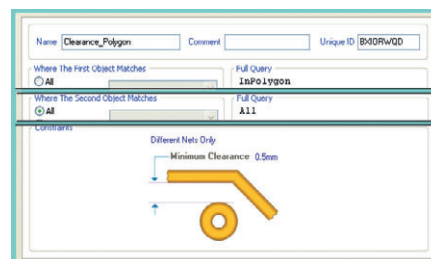


Рис. 7. Правило зазоров для полигона

Пример 2. Общие зазоры на плате в нашем примере составляют 0,25 мм, а для полигонов земли и питания должен быть выдержан зазор 0,5 мм. Создаём новое правило под названием Clearance_Polygon и задаём ему область действия, как показано на рисунке 7. Теперь все полигоны будут отделены на 0,5 мм от остальных объектов на металлизированном слое.

Пример 3. Если на плате имеются корпуса BGA, то под ними невозможно использовать переходные отверстия, заданные по умолчанию для всех цепей. Чаще всего это касается цепей питания, для которых могли быть указаны специфические переходные отверстия, обычно большего диаметра, чем остальные. Таким образом, необходимо указать программе, что под корпусом BGA можно устанавливать только определённые переходные отверстия.

В нашем случае создадим правило, которое задаёт тип переходных

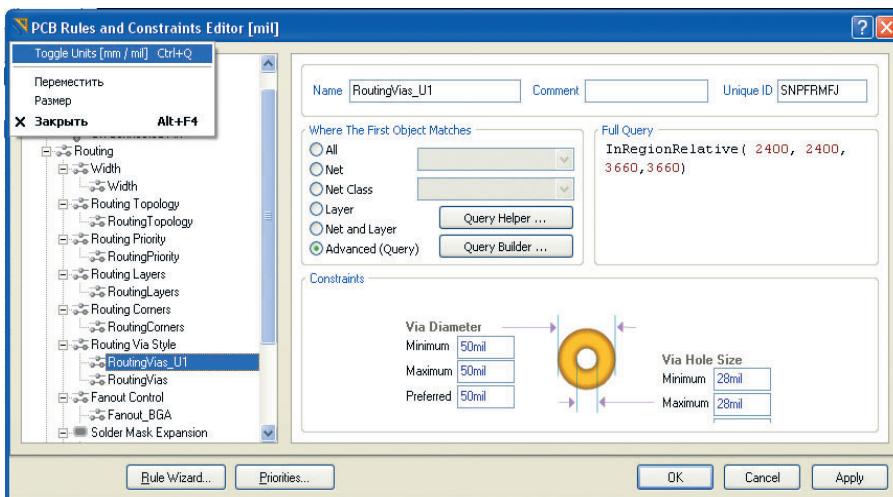


Рис. 8. Правило зазоров для переходных отверстий в заданной области платы

отверстий под корпусом микросхемы U1. Небольшое уточнение: данное правило будет привязано не к корпусу микросхемы, а к участку платы, на котором установлена данная микросхема, т.е. компонент, для которого создаётся данное правило, должен быть установлен заранее.

Итак, создадим новое правило *RoutingVias_U1*, и для выбора области действия выберем параметр *Query Helper*. Далее в появившемся окне (см. рис. 6) выбираем команду *InRegion* из группы *Membership Check*. Чтобы узнать, как описывается данная команда с точки зрения синтаксиса запросов, нажимаем клавишу F1; в появившемся окне будет отображена следующая информация:

InRegion Membership Check
Description
 Returns objects that reside within the region specified by the Query.
Note: The parent objects of group objects (Components, Coordinates, Dimensions, Nets, and Polygons) are not also returned.
 All coordinates are referenced to the absolute origin (and not to the user origin or component insertion point), and at present, all coordinates always use units of mils. (1 mil = 0.001 inch)
Note: This is an alias for *InRegionAbsolute* (Membership Check).
Syntax
InRegion(X1 : Distance_String ,

Y1 : Distance_String , X2 : Distance_String , Y2 : Distance_String) : Boolean
 Each instance of *Distance_String* is a numerical string that specifies a coordinate in the PCB (Library) file. The left and right boundaries of the region are specified by the X1 and X2 parameters, while the lower and upper boundaries of the region are specified by the Y1 and Y2 parameters.
Example
InRegion(12700,4700,13300,5000)
InRegion(12700,4700,13300,5000) = True
 Returns objects residing within the rectangular region bounded by X = 12700 and X = 13300, and by Y = 4700 and Y = 5000; those coordinates are referenced to the absolute origin.


Видно, что требуется задать левую нижнюю и правую верхнюю координаты области платы, причём их значения задаются только в миллах относительно точки привязки. Убедившись, что выбрана дюймовая система координат и начало отсчёта расположено в левом нижнем углу платы, определяем координаты области под компонентом. В нашем случае запрос выглядит, как показано на рисунке 8. После описания области действия необходимо задать значение правила, которое хотелось бы указывать и просматривать в миллиметрах. Можно, не закрывая окна с правилами, переключить единицы измерений на миллиметры, нажав пиктограмму  в левом верхнем углу окна (см.

рис. 8) или комбинацию клавиш CTRL+Q.

Количество правил, которые можно задать с помощью языка запросов, ограничивается только фантазией оператора. Примеры правил можно посмотреть в файлах плат, находящихся в архиве дополнительных материалов к статье на интернет-странице журнала СЭ (www.soel.ru).

Мы рассмотрели способы задания правил проектирования посредством команды *Design > Rules*, но в программе Altium Designer есть альтернативный способ задания правил – помощник Rule Wizard, который запускается посредством *Design > Rule Wizard*. В появившемся окне нажимаем Next, и программа предлагает выбрать тип правила из списка. Выбираем правило *Clearance*, в поле Name вписываем название *Clearance_GND* и нажимаем Next. Далее программа предлагает выбрать область действия правила (см. выше); выбираем 1Net и нажимаем Next. В следующем окне предлагается указать конкретную цепь, – выбираем GND.

Следующий этап – установка приоритета действия правила, после чего программа спрашивает о необходимости занесения правила в список включенных правил и предлагает завершение, – нажимаем Finish. После чего открывается окно PCB Rules, в котором необходимо ввести числовое значение для данного правила.

Таким образом, мы рассмотрели два варианта установки правил проектирования. Первый вариант применим для создания полного списка правил для всего проекта; второй вариант позволяет задать правила поэтапно, что исключает ошибку. Создание правил с помощью мастера подробно не описывается, т.к. выполняется существенно проще «ручного» варианта и обычно не вызывает трудностей у начинающих пользователей.

Более подробно ознакомиться с правилами проектирования можно в файле справки TR0116 Design Rules Reference.pdf, а с особенностями языка запросов – в файле TR0110 Query Language Reference.pdf. В следующей статье будут рассмотрены инструменты размещения компонентов и трассировки проводников на печатной плате.



Создан высокоэффективный двигатель на основе аморфного металла

Hitachi представила прототип двигателя, который использует сердечник, изготовленный из аморфного металла. Этот высокоэффективный мотор не использует магниты из редкоземельных металлов, таких как неодим или диспрозий. Эффективность составила 86% для прототипа 150-Вт двигателя, часто применяемого в кондиционерах. Эффективность существующих образцов с редкоземельными магнитами производства Hitachi достигает 81%, по данным компании.



Такие моторы, как ожидается, будут широко представлены в различных областях, включая промышленное оборудование, потребительскую электронику и автомобили. В настоящий момент на двигателях приходится около половины всей потребляемой энергии, поэтому повышение их эффективности является важной задачей при решении вопроса экономии энергии. Для достижения высоких характеристик используют редкоземельные металлы. В частности, широкое применение нашли Nd-Fe-B-магниты. Для сохранения характеристик при высокой температуре в состав основанных на неодиме магнитов вводят диспрозий.

Однако, несмотря на то что потенциал для увеличения эффективности за счёт применения редкоземельных магнитов сохраняется, рост цен на них становится серьёзной проблемой. «Цены сейчас стали более стабильными после резкого роста в прошлом, но они слишком высоки», — заявляют в Hitachi. Цена одного килограмма неодима и диспрозия выросла втрое до \$30 и впятеро до \$155 соответственно в период с сентября 2004 по сентябрь 2008 г. Цены остаются на высоком уровне из-за того, что 90% всех редкоземельных металлов добывается в Китае. При этом цена имеет тенденцию к росту, поскольку Китай превращается из страны-производителя в страну-потребителя.

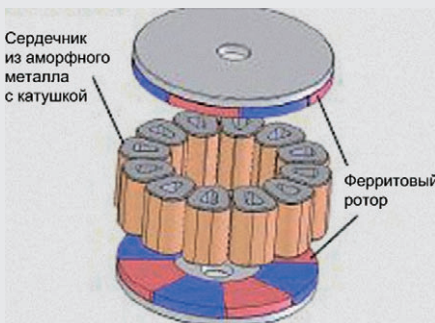
Стальные сердечники с катушками при пропускании по ним электрического тока вращают ротор с постоянными магнитами и генерируют момент на валу двигателя.

В существующих системах используются сердечники, набираемые из листов магнитной стали, а основанные на неодиме магниты применяют в роторе. На этот раз эффективность двигателя была повышена за счёт малых потерь в аморфном металле и новой структуры мотора.

В прототипе мотора сердечник был выполнен из аморфного металла, что позволило использовать ротор с ферритовыми магнитами. Дело в том, что магнитная проницаемость аморфного металла в 10 раз выше, чем у листов электротехнической стали, а потери энергии не превышают десятой доли потерь шихтованной конструкции из стали. При повышении магнитной проницаемости сердечников высокое значение магнитной индукции может быть достигнуто применением слабых магнитов.

Серьёзной проблемой аморфных металлов является сложность в обработке резанием. Листы магнитной стали для сердечников изготавливаются толщиной 0,35...0,50 мм, в то время как толщина аморфного металла составляет 25 мкм.

С другой стороны, Hitachi внесла изменения в структуру и технологию производства сердечников, воспользовавшись опытом производства трансформаторов с аморфными металлами в шихтованных сердечниках. Структура всего двигателя была оптимизирована при помощи трёхмерного магнитного анализа.

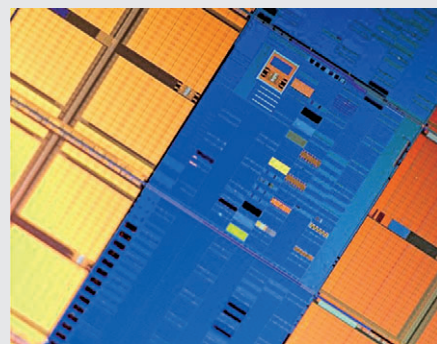


Ближайшая цель разработчика заключается в коммерциализации технологии в течение ближайших трёх лет для использования в промышленном оборудовании. В будущем планируется повысить надёжность и снизить затраты на производство. Также необходимо расширить диапазон мощностей для большего распространения технологии.

techon.nikkeibp.co.jp

Intel раскрывает подробности своих 32-нм микросхем

На конференции IEEE International Electron Devices Meeting (IEDM), по традиции, одним из главных докладчиков станет компания



Intel, представители которой планируют раскрыть детали своего 32-нм технологического процесса для изготовления высокопроизводительных микропроцессоров. Согласно предварительной информации, разработчикам компании удалось создать тестовую интегральную микросхему статической SRAM-памяти ёмкостью 291 Мбит, причём площадь ячейки составляет 0,171 кв. мкм. При этом устройство содержит примерно 2 млрд. транзисторов. Микросхема функционирует на частоте 3,8 ГГц при рабочем напряжении 1,1 В.

Для изготовления интегральных микросхем по 32-нм технологическому процессу разработчики рассчитывают использовать иммерсионную литографию, причём соответствующее оборудование будет приобретаться у японской компании Nikon. Помимо этого используются такие технологии, как технология изготовления затворов с использованием металлических материалов и high-k-материалов, формирование многослойных диэлектрических структур и пр.

Среди остальных докладов есть не менее интересные темы, например, технология изготовления CMOS-микросхем с интегрированными RF-элементами с применением транзисторов на основе фосфида индия, разработанная сотрудниками исследовательской лаборатории HRL Laboratories. Главное преимущество такого подхода — повышение скоростных показателей микросхем по сравнению с традиционными кремниевыми устройствами. Однако их существенным недостатком является сложность в изготовлении и высокая стоимость по сравнению с кремниевыми аналогами.

Интересным обещает быть и доклад сотрудников Университета Тохоку (Tohoku University), в котором будет затронута тема применения элементов на основе магнитного туннельного перехода для создания блоков хранения информации в микропроцессорах с 3D-структурой высокой плотности.

eetimes.com