

# Altium Designer Summer 08 – разработка библиотек и моделей компонентов

Алексей Сабунин (Москва)

В статье приведены основные понятия о библиотеках и моделях компонентов в программе Altium Designer.

Прежде чем приступить к проекту, разработчик должен иметь в своём распоряжении библиотеку компонентов элементной базы, используемой в проекте. Хотя библиотеки, поставляемые с Altium Designer, содержат порядка 86 000 компонентов ([www.altium.com/Community/support/Libraries/Designerlibraries](http://www.altium.com/Community/support/Libraries/Designerlibraries) – описание всех библиотек), существуют сомнения относительно целесообразности их использования. Во-первых, УГО (условные графические обозначения) всех этих компонентов не соответствуют ГОСТ, во-вторых, в этих библиотеках отсутствует российская элементная база, которая применяется на наших предприятиях. Поэтому перед разработкой электрических принципиальных схем и проектированием плат следует рассмотреть порядок разработки библиотек и моделей компонентов, а также разные подходы к их организации.

## КОНЦЕПЦИЯ БИБЛИОТЕК ALTIUM DESIGNER

Компоненты являются основными блоками электронных изделий. При разработке и подготовке к производству проекта каждый компонент нуждается в различных представлениях: логический символ на схеме (УГО), посадочное место на плате (Footprint), описание в формате Spice для моделирования, описание IBIS-модели для анализа целостности сигналов и трёхмерное описание для объёмного представления готовой платы.

Не обязательно наличие всех этих представлений для каждого компонента, но необходимо наличие стартовой точки, которой в Altium Designer является логический символ (УГО). Каждый компонент должен быть определён как минимум названием в схемной библиотеке. Он может содержать выводы и графический

символ в единственном или многосекционном виде и даже иметь альтернативные отображения. Как таковой он может быть размещён в любом схемном проекте. Однако, до тех пор, пока в компонент не добавлены модели, его нельзя применить на практике.

Для однозначного понимания разьясим термины, наиболее часто используемые в среде Altium Designer.

**Компонент:** общее наименование объекта, который может быть применён в проекте.

**Символ:** общее наименование графического представления компонента, подготовленного для размещения на схеме. Символ может содержать графические объекты, которые определяют внешний вид и выводы для электрического подключения.

**Физический компонент** может быть смонтирован на плате.

**Логический символ:** схемное представление физического компонента.

**Часть (секция):** некоторые компоненты, такие как цепочки резисторов или реле, могут быть построены в виде серии отдельных секций (частей), которые, в свою очередь, могут быть размещены на схеме независимо (рассматриваются как многосекционный компонент).

**Модель:** представление компонента, который используется в некоторой практической сфере деятельности.

**Посадочное место:** это наименование используется для модели, которая представляет компонент на заготовке печатной платы. Посадочное место группирует набор контактных площадок (КП) на плате и изображение корпуса компонента и определяет часть платы, требуемую для монтажа и подсоединения физического компонента на плате.

**Библиотека:** файл, содержащий набор компонентов и набор моделей.

**Библиотека моделей:** файл, содержащий набор моделей компонентов.

**Библиотека компонентов:** файл, содержащий набор схемных компонентов.

**Интегрированная библиотека:** файл, содержащий набор схемных компонентов и их ассоциированные модели.

**Библиотека базы данных:** библиотека компонентов, где все символы имеют ссылки, модели связаны и параметрическая информация сохранена в базе данных на основе ODBC (open database connectivity – интерфейс связи с открытыми базами данных), ADO (ActiveX Data Objects – набор компонентов ActiveX, используемых для доступа к БД, поддерживающим спецификацию OLE DB) или в виде таблиц Excel.

На схемной стадии проект является набором компонентов, которые имеют логическую связность (см. рис. 1). Для тестирования или реализации проект нуждается в передаче в другую область проверки функционирования, такую как моделирование, формирование платы, анализ целостности сигналов и т.д.

Каждая область практической реализации требует некоторой информации о компонентах, а также о путях перераспределения этой информации через выводы символа. Некоторая часть этой информации для области реализации размещается в файлах моделей, формат которых является обычно predetermined. Например, модели IBIS, MDL (принцип минимальной длины описания) и SKT. Некоторая информация не содержится в файлах моделей, – например, распределение выводов SPICE и данные списков цепей должны быть определены средствами системы.

Отметим, что модели целостности сигналов IBIS и модели VRML (virtual reality modeling language – язык моделирования виртуальной реальности) или IGES 3D (международный язык обмена графической информацией)

должны быть предварительно импортированы в формат модели Altium Designer. Модели IBIS импортируются непосредственно в диалоге Signal Integrity Model, который открывается при добавлении SI-модели в компонент. VRML- и IGES-модели должны быть импортированы в PCB3Dlib перед тем, как они могут быть добавлены в компонент схемы.

Вся необходимая информация домена содержится внутри схемного компонента, который сохраняется с помощью отдельного интерфейса для каждой добавляемой модели (см. рис. 2). В сущности комплектная модель является комбинацией распределённой информации моделей, сохраняемой в компоненте, а информация о модели домена сохраняется в библиотеке моделей.

В среде Altium Designer существуют четыре типа доступных для использования библиотек.

**Библиотеки моделей** – модели для каждой области сохраняются в «хранилищах», обычно называемых библиотеками моделей. В некоторых областях, таких как SPICE, где обычно одна модель хранится в одном файле, они также рассматриваются как отдельные файлы (\*.MDL, \*.CKT). В других областях модели обычно группируются в библиотечные файлы, соответствующие заданной организации пользователя, такие как посадочные места, сгруппированные в библиотеки пакетного типа (\*.PcbLib).

**Библиотеки символов** содержат схемные компоненты и интерфейс определения их моделей (\*.SchLib). Каждый интерфейс определения модели привязан к своей соответствующей библиотеке моделей.

**Интегрированные библиотеки** – это наборы библиотек символов, которые вместе с их привязанными библиотеками моделей «компилированы» в интегрированную библиотеку (\*.IntLib). Преимущество создания интегрированных библиотек состоит в том, что вся компонентная информация доступна в едином файле. Интегрированные библиотеки не могут быть отредактированы без распаковки источников и перекомпиляции.

**Библиотека базы данных** – библиотека, где все символьные ссылки, привязанные модели и параметрическая информация хранятся в базе данных на основе ODBC, ADO или в формате таблиц Excel. Каждая запись в базе

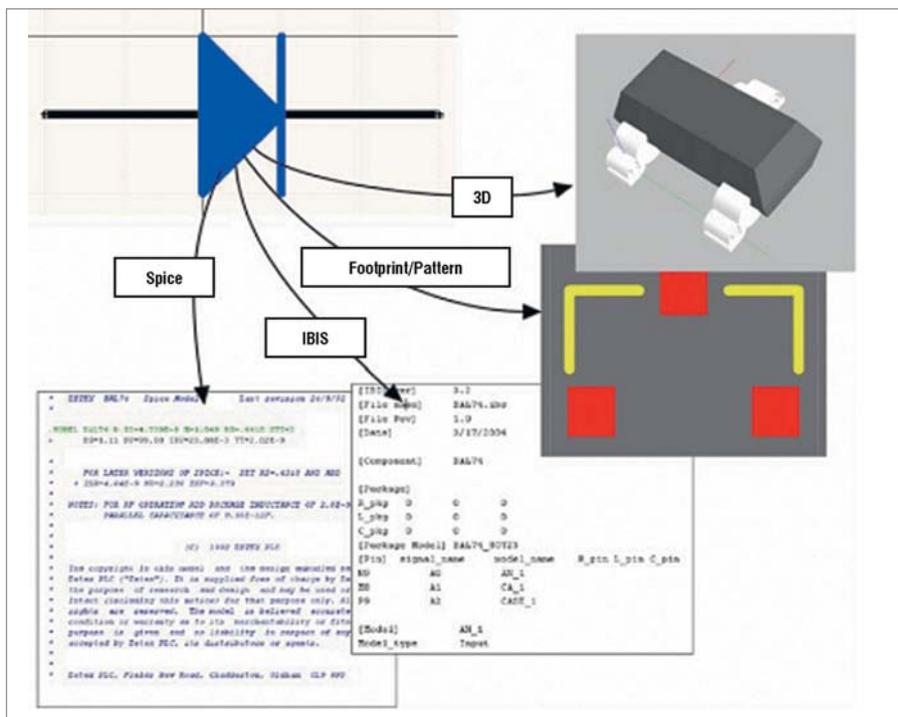


Рис. 1. Информация о различных вариантах представления компонента в файлах моделей

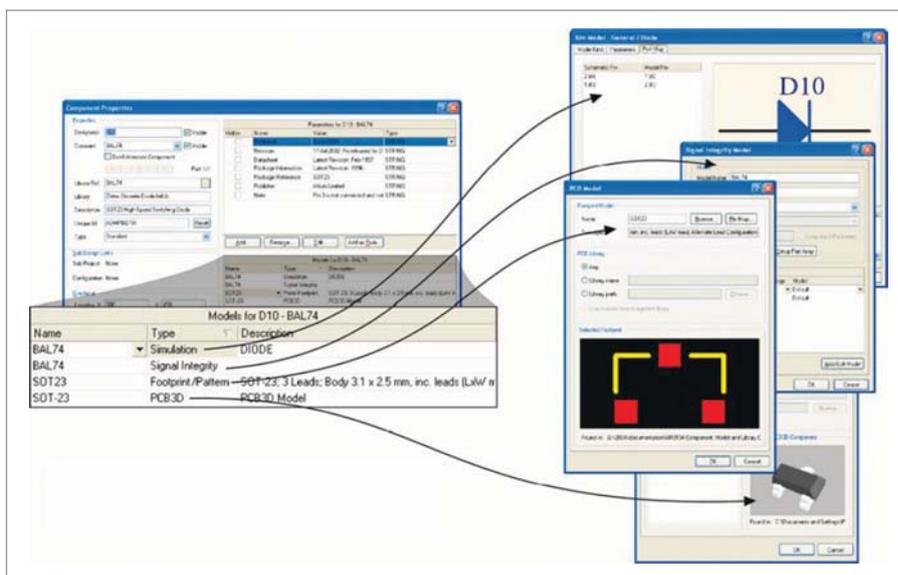


Рис. 2. Связи в каждой модели и любое её переопределение требуют определения модели в соответствующем диалог

данных представляет отдельный компонент, а также все сохраненные параметры, вместе со ссылками на модели. Запись может содержать ссылки на инвентарную ведомость или на другие корпоративные данные о компонентах.

При таком подходе схемные компоненты используются только в виде символов (они не имеют ссылок на модели, описанные в схемной библиотеке) с моделями (посадочными местами или 3D-моделями), сохранёнными в стандартных библиотеках посадочных мест, 3D-компонентов и т.д.

Интерфейсом библиотеки базы данных является документ \*.DBLib, который описывает, какие поля базы данных требуются и какие параметры компонентов они отображают. База данных DBLib подключается через панель Libraries подобно всем другим библиотекам системы.

При выполнении действий по размещению, выполняемых с компонентами библиотеки базы данных, запись в базе данных проверяется, символ загружается из указанной схемной библиотеки и модели добавляются к любым указанным моделям, так же как параметры.

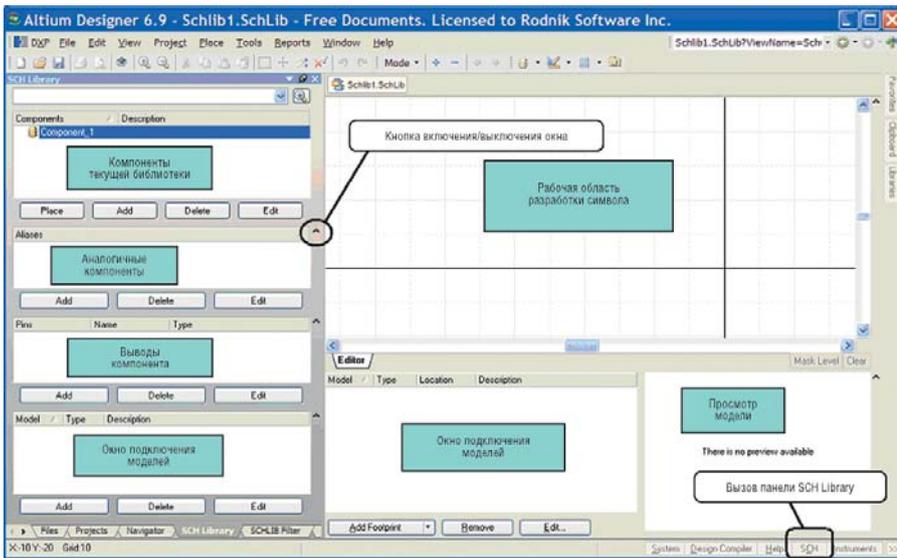


Рис. 3. Интерфейс редактора компонентов

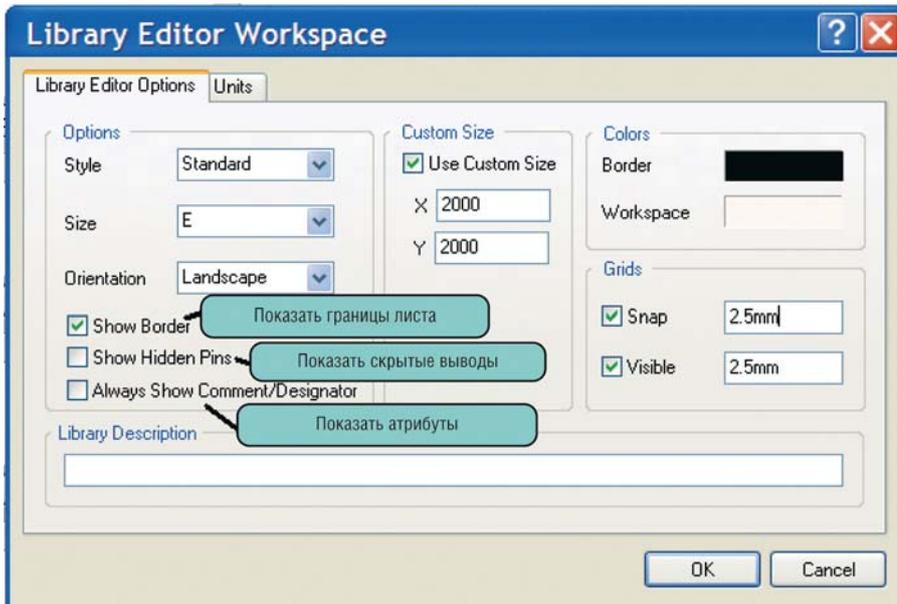


Рис. 4. Настройки рабочей области редактора символов

## РАЗРАБОТКА УГО КОМПОНЕНТОВ

Предполагается, что программа Altium Designer уже запущена на компьютере. Для создания новой библиотеки необходимо выполнить: *File > New > Library > Schematic Library*. Для работы с библиотекой символов

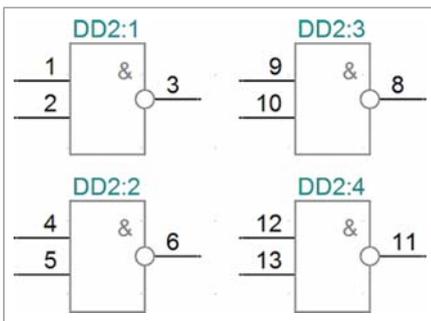


Рис. 5. Пример компонента – микросхема K1554ЛАЗ

необходимо открыть панель (которая вызывается по кнопке *SCH > SCH Library* в нижней левой части окна) и разместить её слева от рабочей области (о размещении панелей см. СЭ № 5, 2008), после чего интерфейс Altium Designer будет иметь вид, показанный на рисунке 3.

Прежде чем приступить к разработке нового символа, необходимо настроить рабочую область, т.е. выбрать необходимые единицы измерения и сетки. Данная настройка производится в окне *Library Editor Workspace*, которое вызывается командой *Tools > Document Options*. В появившемся окне на вкладке *Units* выбираются единицы измерения, причём они могут быть метрические и дюймовые (в нашем случае следует установить миллиметры). На вкладке

*Library Editor Options* следует установить шаг сетки 2,5 мм для двух видов сеток (Snap – сетка перемещения курсора в режиме графической команды, Visible – сетка, отображаемая на экране). Кроме сеток, на данной вкладке могут быть установлены дополнительные параметры (см. рис. 4).

Рассмотрим алгоритм создания компонента на примере микросхемы, изображённой на рисунке 5. Для создания компонента следует нажать кнопку *Add* под списком компонентов данной библиотеки в панели *SCH Library* и в появившемся окне написать название нового компонента. В результате этого новый компонент добавляется в список библиотеки. Далее процесс создания компонента можно разбить на три этапа:

- установка выводов компонента;
- рисование графики символа;
- установка параметров (атрибутов).

1. Выводы компонента устанавливаются командой *Place > Pin*, причём сразу после выполнения этой команды вывод становится привязанным к курсору, и в этот момент необходимо нажать клавишу *Tab* для установки свойств вывода. В открывшемся окне (см. рис. 6) следует указать название и позиционное обозначение вывода, которое соответствует номеру вывода корпуса микросхемы. Обязательно следует указать длину вывода, кратную 2,5 мм; остальные параметры, такие как электрический тип вывода и специальные символы, добавляются позже. При установке вывода он привязан к курсору большим вертикальным крестом, с другой стороны маленький диагональный крестик показывает электрическое окончание вывода (см. рис. 7). Для размещения первого вывода данного компонента его следует развернуть на 180 градусов нажатием клавиши *Space* (пробел). Установка вывода производится нажатием левой кнопки мыши, причём после установки в электрическом окончании вывода показаны четыре белые точки. Установим три вывода, как показано на рисунке 5, причём у вывода 3 следует отобразить инверсию, для чего в свойствах компонента параметру *Outside Edge* (символ снаружи на границе контура) выбирается значение *Dot* (Знак инверсии).

2. На втором этапе создаётся графика УГО, для чего используются коман-

ды рисования (*line, Ellipses, Arc* и др.). В нашем случае выбираем команду *Place > Line* и рисуем прямоугольник, как показано на рисунке 5. Толщина и свойства рисуемой линии могут быть изменены во время рисования при нажатии клавиши *Tab*, причём толщина линии задаётся условно *Smallest, Small, Medium* и *Large* (*Small* соответствует толщине 2,54 мм). При рисовании линии могут быть использованы пять режимов, переключение между которыми выполняется клавишей *Space* (Пробел). Во время рисования графики символа может возникнуть необходимость переключения шага сетки. Для переключения между стандартными режимами сеток (установка которых будет рассмотрена в следующей статье) необходимо нажать клавишу *G* (при этом сетка будет циклически переключаться между значениями 1, 2,5 и 5 мм). При необходимости включить шаг сетки, отличный от указанных выше, нажмите последовательно клавиши *V* и *G*, после чего в появившемся выпадающем меню выберите команду *Set Snap Grid*.

Создав первую ячейку микросхемы, необходимо повторить первые два шага и создать остальные три ячейки, при этом на символе позиционные обозначения выводов должны совпадать с реальными номерами выводов микросхемы. Для создания новой ячейки внутри одного компонента выполняется команда *Tools > New Part*. В рассматриваемом случае выводы 7 и 14 относятся к питанию и земле и не должны быть показаны на символе. Для установки скрытых выводов необходимо (командой *Place > Pin*) зайти в свойства компонента нажатием клавиши *Tab*. В свойствах компонента после установки соответствующего обозначения и названия следует включить опцию *Hide* и в поле *Connect To* задать название цепи, к которой по умолчанию будет подключаться данный вывод. Кроме этого, для таких выводов обычно ставят значение 0 в поле *Part Number*, что говорит о принадлежности данного вывода к какой-то конкретной ячейке микросхемы. Результат создания символа показан на рисунке 8.

3. При создании символа в программе PCAD после вышеописанных шагов требовалось указать точку

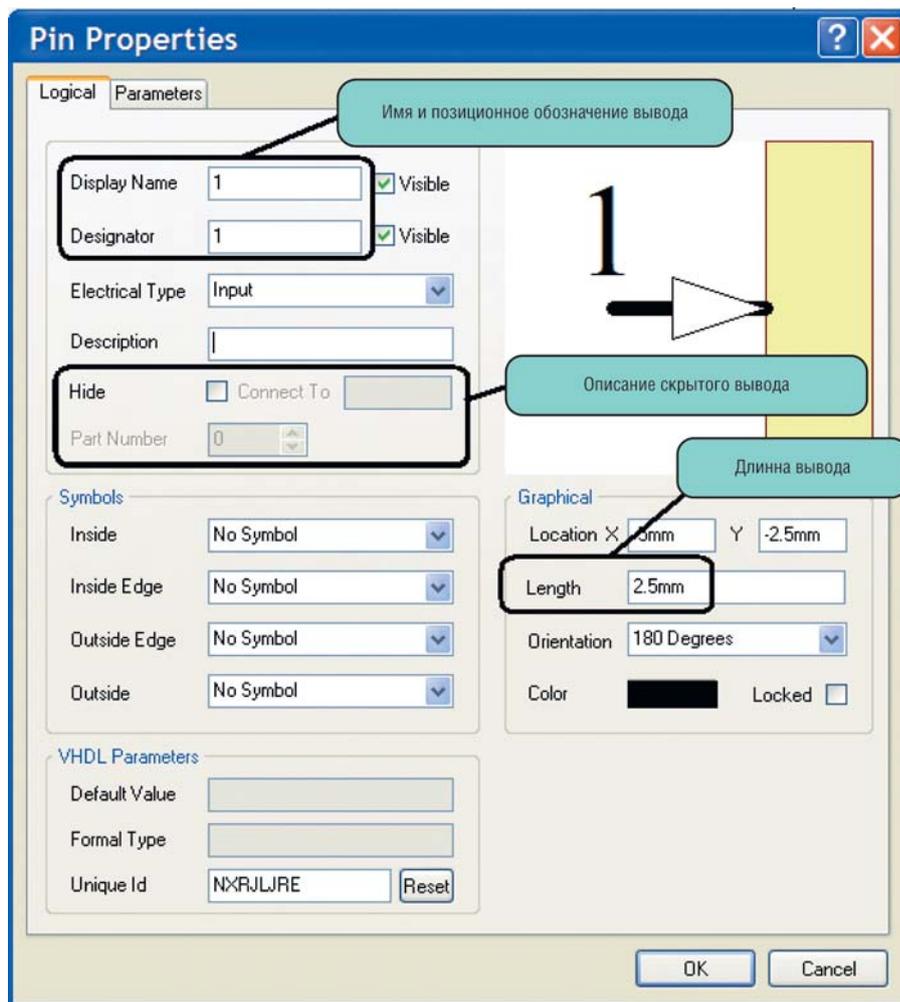


Рис. 6. Свойства вывода компонента

привязки и установить обязательный атрибут *RefDes*. В программе Altium Designer точка привязки определяется началом координат рабочей области (перекрестие в центре экрана), а позиционное обозначение (*RefDes*) автоматически добавляется над корпусом микросхемы. Третьим этапом при создании символа является добавление атрибутов, которые в дальнейшем могут быть использованы для оформления текстовой документации (перечней и спецификаций). В программе PCAD этот этап выполнялся последним, при создании компонента в приложении *Library Executive*. В программе Altium Designer в связи с тем, что изменена структура библиотек, компонент создаётся уже сейчас! Для установки атрибутов компонента необходимо выполнить двойной щелчок мыши на его названии в панели *SCH Library*. В появившемся окне свойств компонента (см. рис. 9) необходимо указать префикс позиционного обозначения компонента и при необходимости добавить па-

раметры, которые в дальнейшем могут быть использованы для создания спецификации и перечня элементов. После нажатия кнопки *OK* можно выполнить сохранение библиотеки нажатием пиктограммы с изображением дискеты.

### РАЗРАБОТКА ПОСАДОЧНЫХ МЕСТ (FOOTPRINT) КОМПОНЕНТОВ

При реализации баз компонентов в программе PCAD использовались библиотеки одного типа (с расширением *.lib*). Эти библиотеки можно было назвать интегрированными, т.к. внутри них хранились символы, посадочные места и сами компоненты. В программе Altium Designer, как было показано выше, для каждого типа объектов имеются свои библиотеки. При этом библиотеки символов по



Рис. 7. Установка вывода

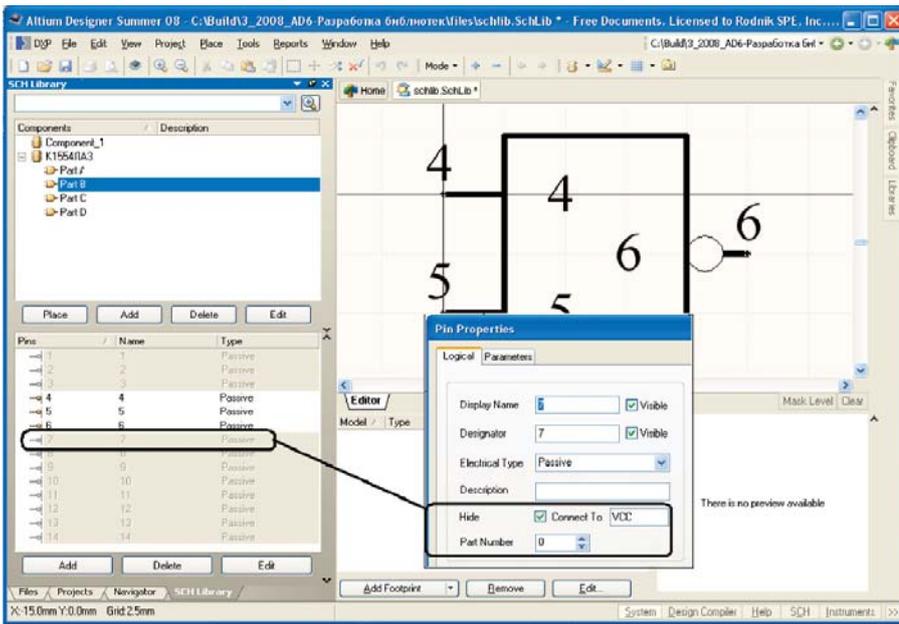


Рис. 8. Установка скрытых выводов

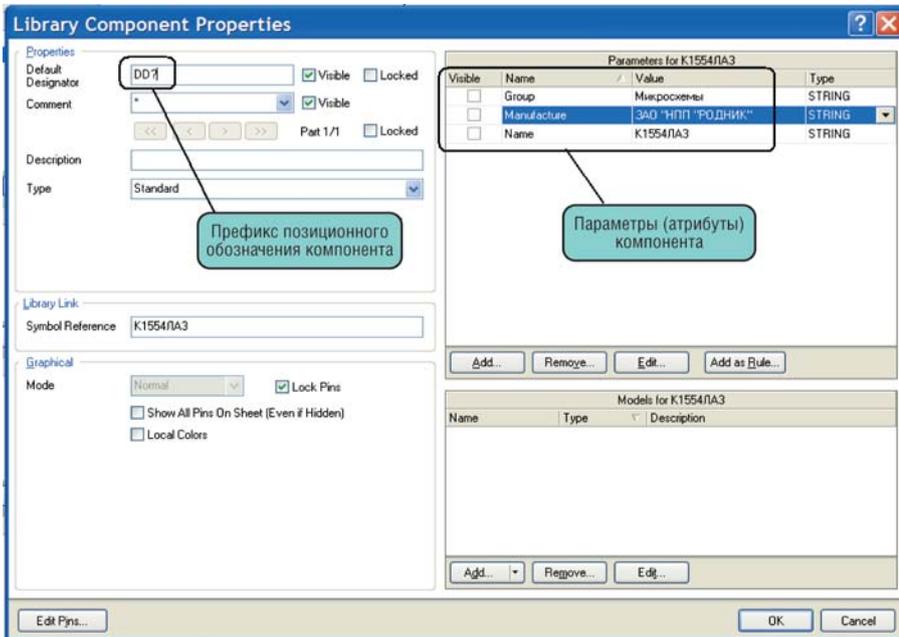


Рис. 9. Свойства компонента

сти представляют собой библиотеки компонентов, однако библиотеки посадочных мест имеют отдельное расширение; остальные три типа моделей (3D, Spice, IBIS) не имеют своего типа библиотек и хранятся в виде отдельных файлов.

Итак, после создания компонента в виде символа микросхемы K1554PA3 приступим к созданию посадочного места этой микросхемы. Начнём с создания новой библиотеки, для чего выполним команду *File > New > Library > PCB Library*.

Для работы с редактором посадочных мест используется панель *PCB Library*, которая может не запускаться автоматически. Для отображения

указанной панели необходимо найти её в группе панелей *PCB* в левой нижней части экрана. После этого интерфейс программы примет вид, показанный на рисунке 10, причём нетрудно заметить, что по умолчанию новый документ запускается в милах (mil). Такие операции, как настройка рабочей области, сетки, свойства линии и других графических объектов «по умолчанию» мы будем рассматривать в следующей статье; в данном случае установим параметры только для текущей библиотеки.

Для установки параметров выполним команду *Tools > Library Options*, и на экране появится окно, показанное на рисунке 11, где выставляются едини-

цы измерения (в поле *Units*), шаг сетки (*Snap Grid*), шаг сетки для установки компонентов (*Component Grid*, для библиотек эта сетка не используется) и две видимые сетки (*Grid 1* и *Grid 2*). Видимые сетки задаются относительно сетки *Snap Grid*, первая должна быть равна шагу сетки, вторая – в десять раз больше, тогда настройки сеток Altium Designer будут идентичны настройкам сеток PCAD. Остальные параметры в данном окне пока несущественны, поэтому принимаем изменения нажатием кнопки *OK*. Теперь на экране отображается светлая сетка, а при увеличении масштаба изображения появится тёмная сетка.

Приступим непосредственно к созданию посадочного места, которое состоит из двух этапов:

- установка контактных площадок;
- разработка графического изображения корпуса компонента.

Создание посадочного места может быть выполнено двумя способами: вручную и с помощью мастера. Сначала рассмотрим «ручной» вариант на примере посадочного места резистора, а затем с помощью мастера реализуем посадочное место для описанной выше микросхемы.

Для создания нового посадочного места выполним команду *Tools > New Blank Component*, после чего в панели *PCB Library* появится новое посадочное место с условным названием. Чтобы задать необходимое название создаваемому корпусу, необходимо выполнить двойной щелчок мыши на его надписи в панели *PCB Library* (в нашем случае вводим название *Resistor*).

1. Для установки контактных площадок выполним команду *Place > Pad*, и нажмём клавишу *Tab* для описания параметров контактной площадки. В появившемся окне (см. рис. 12) необходимо задать параметры отверстия (в поле *Hole Information*) и параметры контактной площадки (в поле *Size and Shape*). Кроме этого, могут быть заданы дополнительные параметры (см. рис. 12), причём для площадки под поверхностный монтаж сначала необходимо указать слой в поле *Layer*, после чего поле *Hole Information* будет недоступно. Нумерацию контактных площадок следует начинать с единицы, т.к. здесь указываются номера физических выводов микросхемы. Нулевой номер контактной площадки обычно используется для крепежных отверстий. Для резис-

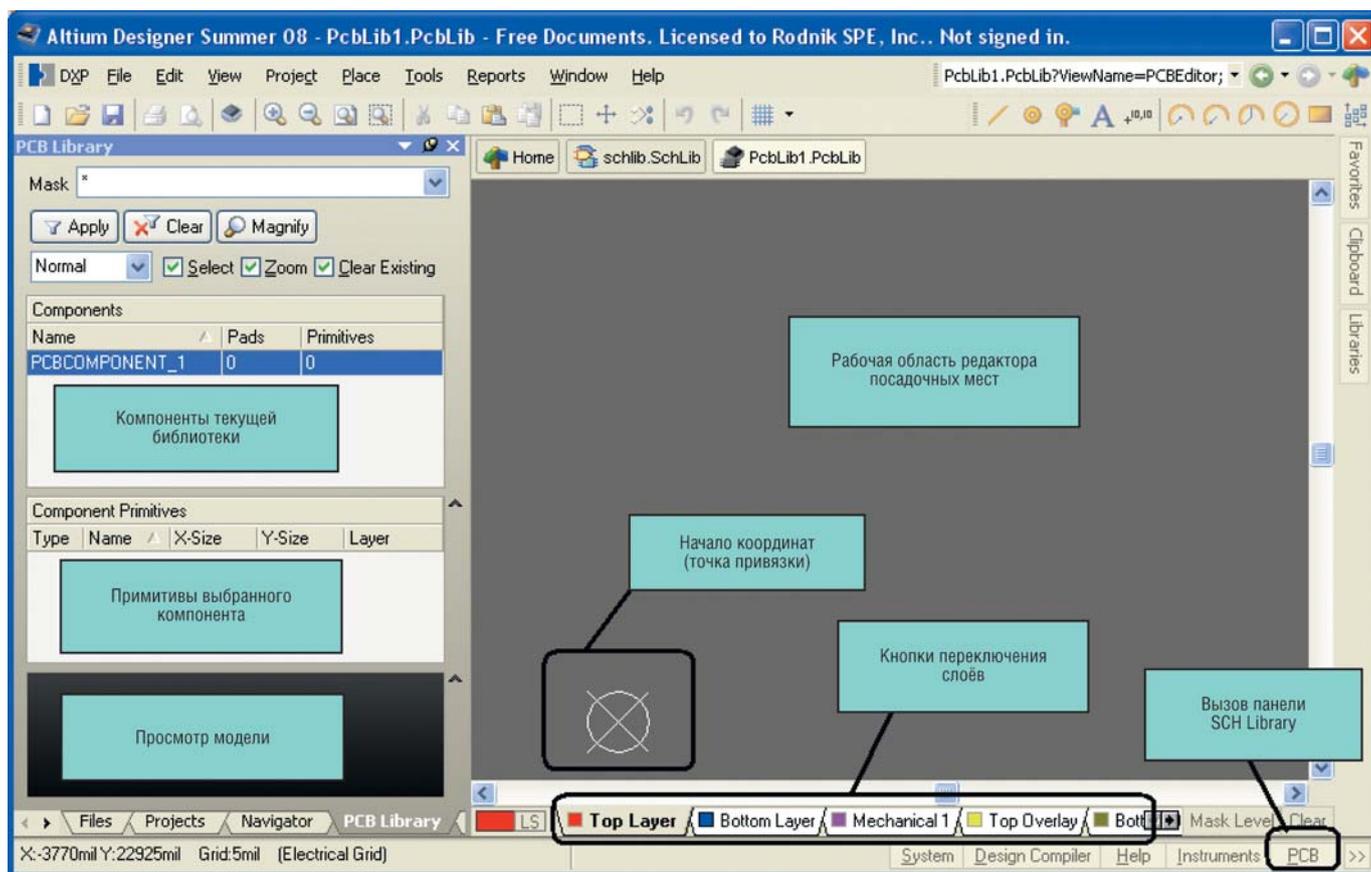


Рис. 10. Интерфейс редактора посадочных мест

тора устанавливаем две контактные площадки, причем первую – в начало координат, а вторую на 10 мм правее (4 шага основной сетки).

2. Вторым этапом создания посадочного места является разработка графики, причём она должна быть выполнена в определённом слое. Более подробно работа со слоями в программе Altium Designer будет рассмотрена в одной из последующих статей. В нашем случае контур корпуса компонента должен быть выполнен в слое *Top Overlay*. Выбор слоёв производится кнопками с их названиями в нижней части экрана (см. рис. 10). Теперь в выбранном слое будем создавать контур компонента с помощью команды *Place > Line*, которая имеет некоторые отличия от аналогичной команды, используемой в редакторе символов. Во-первых, переключение сеток при нажатии клавиши *G* здесь происходит не между заранее заданными значениями, а путём выбора сетки из списка (можно задать пользовательское значение *Set Snap Grid*). Во-вторых, вместо пяти режимов рисования каждый режим разбит на подрежимы, как показано на рисунке 13. Режимы переключаются

комбинацией клавиш *Shift + Space* и отличаются ортогональностью. Подрежимы переключаются клавишей *Shift* и отличаются положением начального сегмента. После завершения рисования корпуса резистора изображение на экране должно соответствовать рисунку 14.

В качестве особенностей данного редактора (по сравнению с PCAD

Pattern Editor) необходимо отметить следующие:

- не ставится атрибут *RefDes*; здесь, как и в редакторе символов, он будет установлен автоматически над корпусом компонента;
- не задаётся точка привязки, и отсутствуют элементы *Glue Point* (точка приклейки), *Pick Point* (точка захвата для автоматической установки)

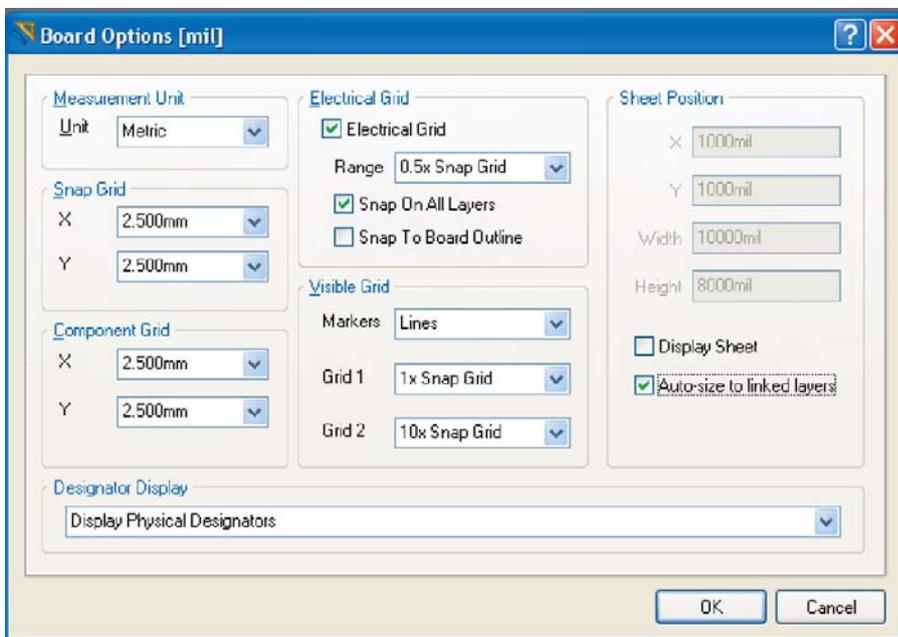


Рис. 11. Настройки рабочей области редактора посадочных мест

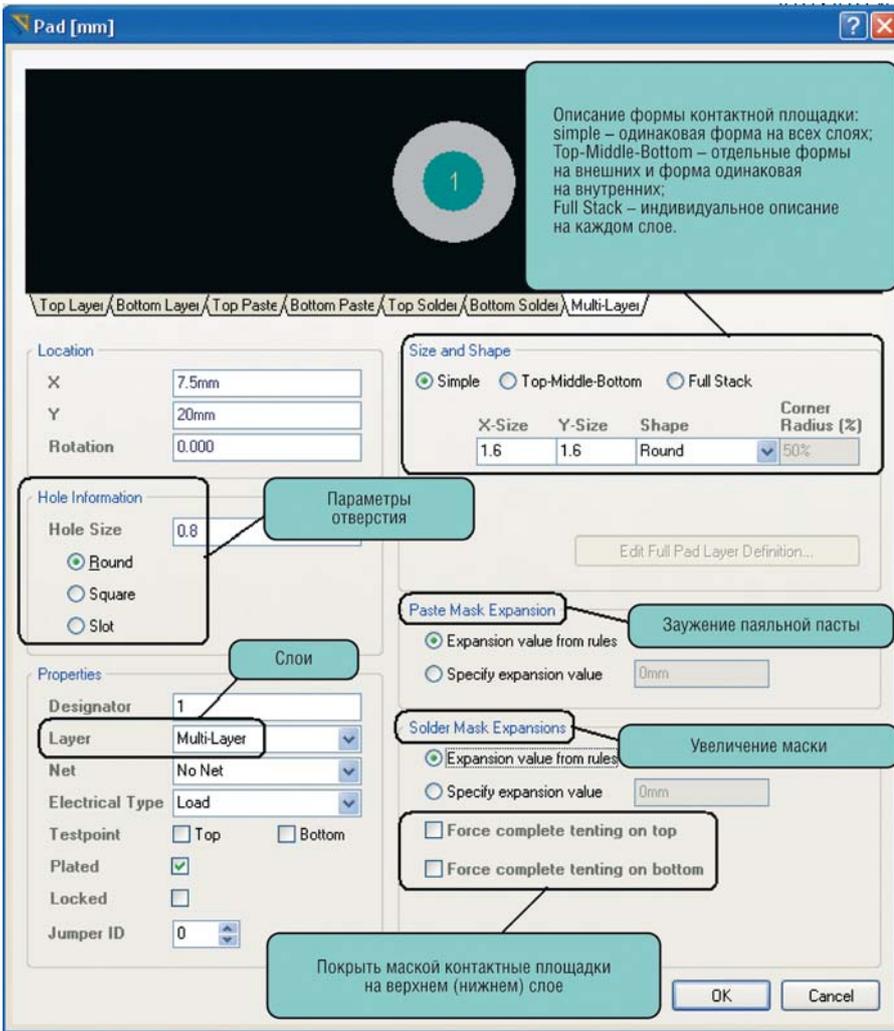


Рис. 12. Параметры контактной площадки

и *Test Point* (тестовая точка). Точка привязки по умолчанию задана в начале координат, а точки *Glue Point* и *Pick Point* с ней совпадают, поэтому, если возникает необходимость в их использовании, следует располагать начало координат в геометрическом центре компонента. Данную операцию удобно выполнять по окончании разработки посадочного места (с помощью команды *Edit>SetReference>Center*).

Когда разработка посадочного места резистора закончена, покажем, как создать более сложное посадочное место микросхемы в режиме мастера. Для запуска мастера выполним команду *Tools > Component Wizard*. На экране появится диалоговое окно мастера создания посадочных мест, и после нажатия кнопки *Next* будет предложено выбрать тип корпуса и единицы измерения (см. рис. 15). Далее, в пошаговом режиме будет предложено задать набор параметров, которые описывают создаваемую модель, – например, параметры контактной площадки, число выводов,

толщину линии корпуса и т.д. В нашем случае создаём корпус с 14 выводами. На последнем шаге будет предложено указать название посадочного места.

После завершения работы мастера сохраним библиотеку нажатием пиктограммы с изображением дискеты. Теперь мы имеем в своём распоряжении две библиотеки: в одной хранится символ микросхемы, а в другой – посадочные места резистора и микросхемы. Последним этапом нашей работы, по результатам которого можно будет использовать полнофункциональный компонент на схеме и плате, является установка ссылки для компонента на модель посадочного места. Данная процедура может быть выполнена различными способами, и существуют несколько вариантов реализации библиотек. В рассматриваемом примере будет показан метод, который предполагает хранение всех компонентов и моделей в виде интегрированных библиотек.

Для создания интегрированной библиотеки выполним команду *File>*

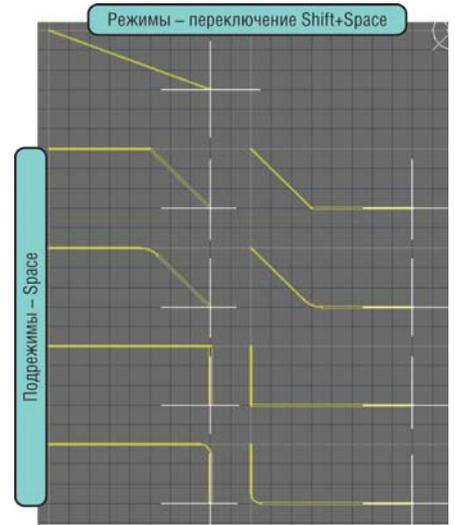


Рис. 13. Режимы рисования линии

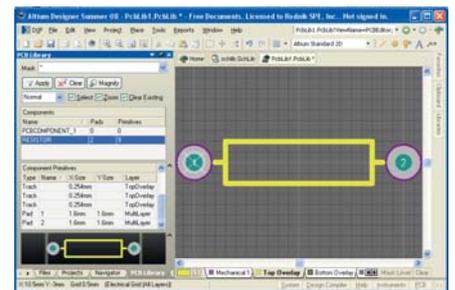


Рис. 14. Готовое посадочное место

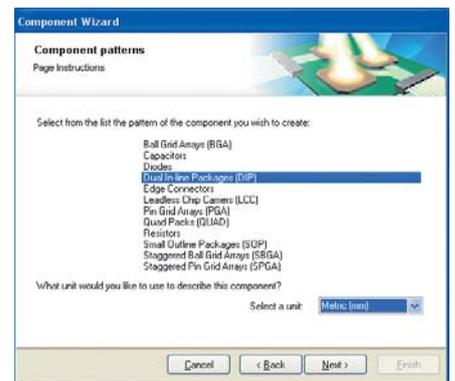


Рис. 15. Мастер создания посадочных мест

*> New>Project>Integrated Library*, после чего в структуре панели *Project* добавится новый документ, который необходимо сохранить, нажав на нём правой кнопкой мыши и выполнив команду *Save Project*. Теперь необходимо добавить в структуру созданного проекта ранее созданные библиотеки путём их перемещения в дерево панели *Project* (см. рис. 16).

Когда все необходимые библиотеки находятся в структуре интегрированной библиотеки, можно выполнять подключение моделей к соответствующим компонентам. В нашем случае необходимо открыть библиотеку символов и нажать кнопку *Add Footprint* в окне подключения моде-

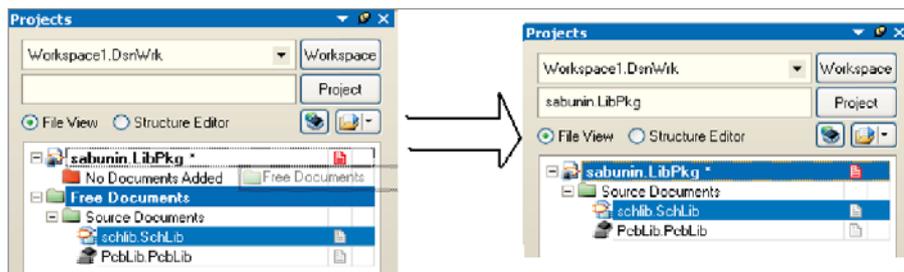


Рис. 16. Создание интегрированной библиотеки

лей (см. рис. 3). В результате на экране появится окно *PCB Model* (см. рис. 17), где после нажатия кнопки *Browse* необходимо указать требуемый корпус. Причём посадочные места, доступные по кнопке *Browse*, сразу имеют ссылку на библиотеку посадочных мест, которая находится в интегрированной библиотеке.

Теперь мы имеем готовый компонент K1554JA3, но прежде чем его использовать, необходимо сделать проверку. Она может быть выполнена по отдельности на каждом этапе разработки (символ, посадочное место и т.д.), или можно выполнить полную проверку библиотеки компонентов. Проверки первого типа выполняются командой *Reports > Component Rule Check*, как в редакторе символов, так и в редакторе посадочных мест, и фиксируют простейшие ошибки: наличие дублирующихся выводов, атрибутов и т.д.

Общая проверка библиотеки компонентов выполняется компиляцией, по результатам которой будет сгенерирован файл \*.IntLib, который можно в дальнейшем использовать для работы при создании схем. Итак, выполним компиляцию проекта *Project > Compile Integrated Library...*, по результатам которой интегрированная библиотека будет автоматически подключена к программе и готова к использованию. В нашем случае мы не стали задавать настройки компиляции, т.е. сделали проверку библиотеки по правилам, заданным «по умолчанию» и, соответственно, избежали каких-либо ошибок в проекте. Компиляция и работа над её ошибками будут рассмотрены подробно в одной из последующих статей.

В заключение следует отметить, что процесс разработки библиотек в программе Altium Designer значительно упрощён (по сравнению с PCAD) за счёт исключения этапа заполнения таблицы соответствия выводов, но при этом немного услож-

нён этап разработки символа, поскольку теперь необходимо рисовать по отдельности каждую ячейку сложной микросхемы. В данной статье приведена самая предварительная информация о разработке библиотек и ведении баз данных в Altium Designer. Некоторые вопросы будут рассмотрены в последующих публикациях. Более подробно с ней можно ознакомиться в справочном руководстве. В папке *Help* установочной директории Altium Designer этой теме посвящены документы AR0104, TU0103 и AP0134.

В следующей статье будет описан процесс разработки электрических принципиальных схем и компиляция проекта.

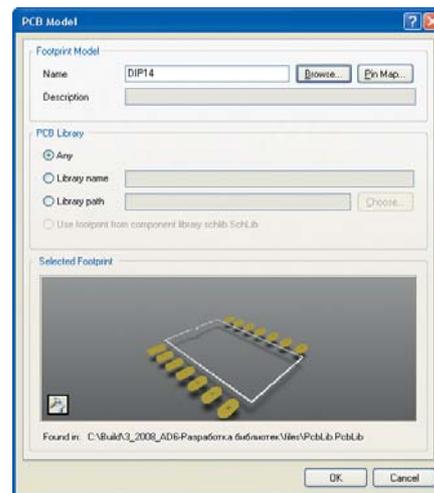


Рис. 17. Подключение посадочного места

В мае 2008 г. австралийская компания Altium выпустила новую версию программы под названием Altium Designer Summer 08. В данном цикле статей рассматриваются основные этапы работы с программой, которые не были изменены при выпуске новой версии. Ознакомиться с новыми возможностями пакета Altium Designer Summer 08 можно на интернет-странице [www.altium.com/summer08](http://www.altium.com/summer08). ©