

Создание библиотечного компонента в P-CAD 2001 Library Executive

Украинский А.С., aukrainsky@granch.ru

Эти сугубо практические рекомендации написаны в первую очередь для того, чтобы помочь читателю сделать первый шаг и убедиться, что создание своих библиотек - вещь действительно очень простая. Приведен минимальный, но достаточный набор сведений, в надежде, что читатель, при необходимости, сам разберется с остальным, используя встроенную в пакет помощь и другие литературные источники.

Для начала определимся с терминологией.

1. Понятия, относящиеся к выводам ЭРЭ.

Все сказанное ниже относится к P-CAD 2001 и найдено "опытным путем". Даже в ACCEL EDA v.15.xx.xx, до переименования его в P-CAD, часть этих понятий трактовалась несколько иначе!

— Pin Num (Number) – номер вывода. Задается автоматически при рисовании символа ЭРЭ (в порядке нанесения выводов, с управляемым исходным значением и приращением), *на изображении символа и схемах никогда не виден*. Виден в Properties (свойствах) вывода, а также в таблице Pin View, при описании упаковки элементов в корпус. Используется для указания логической эквивалентности выводов как в пределах элемента, так и у одинаковых элементов, упакованных в один корпус. Может быть изменен в меню Properties (свойства) вывода, либо через утилиту Renumbr.

— Pin Des (Designator, описание). Может быть как цифровым, *так и текстовым*. Задается автоматически при рисовании символа ЭРЭ (в порядке нанесения выводов, с управляемым исходным значением и приращением). На изображении символа и схемах виден там, где мы обычно наносим реальный номер вывода, т.е. у вывода, вне контура символа! Может быть переназначен для каждого из элементов, входящих в один корпус, в таблице Pin View, при описании упаковки элементов в корпус. Может нести любую, необходимую разработчику схемы смысловую нагрузку.

Пока это несколько непривычно, поэтому: при создании символа рекомендую задавать Pin Des, повторяя Pin Num. Затем, при описании упаковки элементов в корпус (в таблице Pin View), для каждого вывода Pin Des вручную должен быть изменен в соответствии с реальной цоколевкой ЭРЭ.

— Pin Name – имя вывода. На изображении символа и схемах виден у вывода, внутри контура символа! Вводится при необходимости, персонально для каждого вывода, при создании символа.

— Pad Number – номер штырька на корпусе, соответствующего выводу. В традиционном понимании, понятия Pad Number и Pin Des были эквивалентны. Сейчас их разделили. Именно Pad Number используется при автоматическом создании таблиц связи для разводчика печатных плат. На электрических схемах не виден нигде! Появляется в первый раз в таблице Pin View, при описании упаковки элементов в корпус.

Исходя из принятого нами раньше решения использовать при задании Pin Des реальную цоколевку ЭРЭ, значения Pad Number и Pin Des должны совпадать!

2. "Символ" – это условное графическое обозначение (УГО) электрорадиоэлемента (ЭРЭ), выполненное (желательно) по ЕСКД, и создаваемое с помощью редактора символов – программы Symbol Editor из рассматриваемого нами пакета P-CAD 2001 (ACCEL EDA v.16.01.06).

"Символ" содержит собственно графический рисунок УГО, так называемые атрибуты, а также Place Ref Point - точку привязки УГО к будущему чертежу электрической схемы.

Атрибуты, это служебные тексты, содержащие информацию о ЭРЭ, которая впоследствии может быть использована программами и утилитами пакета ACCEL EDA при рисовании схем, например – для автоматического присвоения позиционного обозначения, при составлении спецификации и т.д.

Атрибуты я условно разделил на обязательные, не очень обязательные и совсем не обязательные.

Я выделил в "обязательные" те атрибуты, без которых (если они не заданы) Symbol Editor *откажется* сохранять созданный символ. "Обязательным" является единственный атрибут – RefDes, служебное поле, в которое при рисовании схемы вносится позиционное обозначение ЭРЭ.

Примечание. Symbol Editor *также откажется* сохранять созданный символ, если для каждого из введенных выводов не прописаны номера Pin Num и Pin Def - условная нумерация выводов в пределах символа. (О переходе от условной нумерации выводов к реальной – см. ниже).

"Не очень обязательный" атрибут - "Type" (тип элемента). В это поле при рисовании схемы автоматически заносится набор знаков, использованный в качестве имени компонента в библиотеке. О том, что он не задан, Symbol Editor с сожалением *известит* Вас при сохранении, но все-таки выполнит ваше желание - сохранит символ без задания его "типа".

Отсутствие остальных атрибутов Symbol Editor не замечает, поэтому и нам поначалу можно без них обойтись.

"Символ" может содержать также произвольные текстовые вставки, с помощью которых я наношу функциональное назначение элемента, например – "&" и его сокращенное наименование, например – "ЛАЗ". (Что существенно упрощает чтение и изучение схемы).

Ввод текста во всех компонентах пакета производится с использованием понятия "Стиль", с которым настоятельно рекомендую разобраться отдельно и основательно. Дело в том, что любому

текстовому фрагменту символа или схемы может быть назначен любой установленный в WINDOWS шрифт, и вдобавок, к стандартной высоте, ширине, толщине элементов шрифта могут быть добавлены "пикадовские" поправки.

Ваши схемы будут выглядеть гораздо опрятнее, если, начиная с создания первых символов, Вы выберете (создадите) для себя и будете использовать не более двух – трех, от силы – четырех стилей текста. Навести порядок в уже созданном – задача очень сложная.

Кстати, отсутствие единообразия в стилях делает практически невозможным использование чужих библиотек.

Само рисование символа производится теми же способами, что и рисование схемы. Меню с инструментами для рисования и нанесения атрибутов и текстов выполнено в виде кнопок, расположенных по умолчанию вдоль левого края экрана.

3. Понятие "компонент" отражает тот факт, что в одном корпусе микросхемы 555ЛА3 содержится четыре одинаковых логических элемента "2И-НЕ".

Видимо, для единообразия, понятие "компонент" применяют и к одноэлементным ЭРЭ – резисторам, конденсаторам и т.п.

Описание компонента содержит:

- буквенный код, используемый в позиционном обозначении компонента (DD, DA, R, C) – "Refdes Prefix",
- количество и перечисление типов входящих в него элементов - "Gate",
- описание факта однотипности входящих в компонент элементов - "Gate Eq",
- подключение к реальным выводам корпуса ЭРЭ (Pad number) условных входов и выходов символов элементов (Pin Des и Pin Num), а также цепей питания.

Компонент может содержать как однородные (homogeneous), так и неоднородные (heterogeneous) элементы. Неоднородные элементы описаны разными символами. Пример неоднородного элемента – на одном из логических элементов микросхемы 590КН5, кроме аналоговых и логических выводов, показаны выводы питания.

4. Программа Symbol Editor обеспечивает создание символа со всеми атрибутами, а так же создание компонента, который я бы назвал первичным. Этот компонент содержит только один элемент - "Gate", который имеет условные номера выводов и который имеет стандартный код U в качестве Refdes Prefix. Для задания требуемых значений указанных сведений, компонент должен быть откорректирован с помощью инструментов программы Library Executive.

5. Программа Symbol Editor может сохранять созданный символ в двух форматах.

Командой "Save To File / Save To File As..." ("Сохранить в файл / Сохранить в файл как...") символ сохраняется в индивидуальный файл с назначенным Вами именем и расширением .sym. При открытии такого файла, рабочее окно редактора приобретает вид, полностью повторяющий момент сохранения, и что самое главное – содержит все использованные Вами сетки.

Командой "Save / Save As..." ("Сохранить / Сохранить как...") символ упаковывается в выбранную Вами библиотеку.

"Библиотека" – файл с расширением *.lib, в котором сохранены как символы элементов, так и компоненты на их основе. В одну библиотеку, как правило, помещают несколько близких по какому либо признаку (например, одной серии) компонентов. Именно из файлов *.lib компоненты доступны программе рисования схем - PCAD Schematic. Файл *.lib заархивирован, т.е. упакован. Плата за это – при вызове на редактирование библиотечного символа, тот, хоть убей, выводится в дюймовой сетке.

6. Памятка написана в предположении, что читатель знаком с WINDOWS, имеет начальные навыки работы с PCAD (ACCEL) Schematic, и в последней работает с использованием метрических размерных сеток (Grids).

Работать с программами Symbol Editor и Schematic гораздо удобнее, если использовать шаблоны – заготовки с необходимой основной надписью, набором сеток и текстовых стилей. Мои шаблоны приведены в приложении к данной памятке. В шаблонах схем использованы "форматы" – рамки и основные надписи по ЕСКД. Они, в свою очередь, нарисованы вручную тем – же редактором Schematic, вынужденно сохранены как схемы (с расширением *.sch), а затем переименованы в *.ttl. К схеме шаблона "форматы" подключены через меню "Options, Configure, Title Sheets". Чтобы мои схемы читались нормально, Вам необходимо скопировать эти файлы на свой рабочий диск и аналогично настроить конфигурацию.

7. Рекомендуемая методика освоения изложенной ниже информации:

а) открыть данный документ (что Вы, как я понимаю, уже сделали!);

б) по мере чтения текста, при появлении рекомендаций типа "Запускаем Library Executive ", выполнять их, не закрывая данный документ, а открывая нужные программы в новом окне;

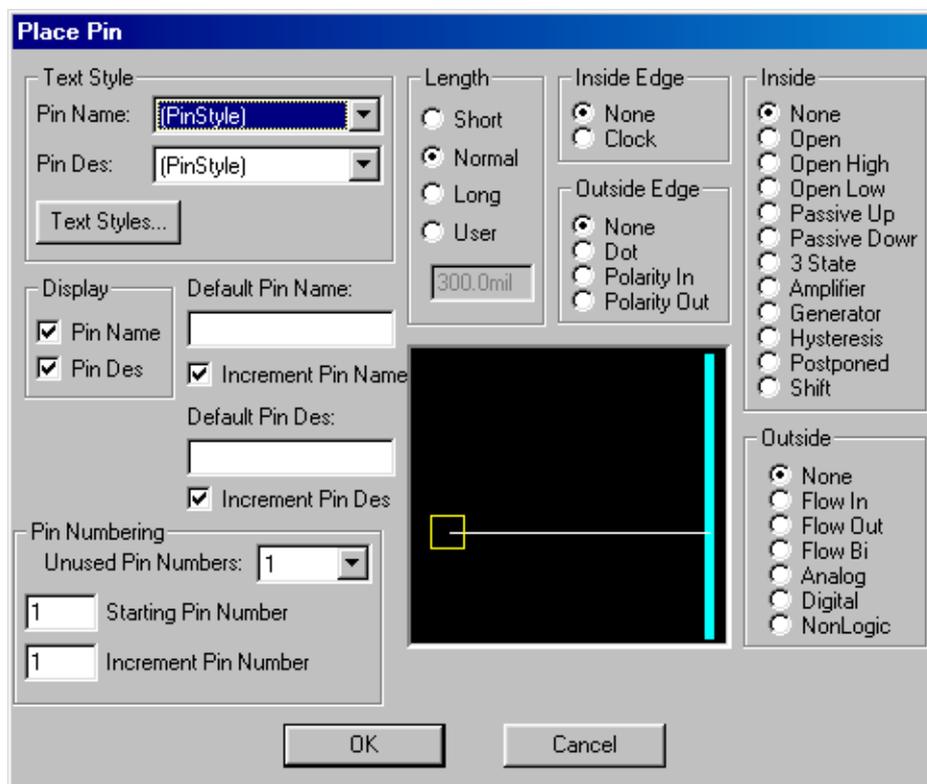
в) двигаться далее, перескакивая из программы в текст и обратно по мере необходимости.

Итак, жизнь вас заставила создать собственный библиотечный элемент?

Тогда поехали!

1. Запускаем PCAD Library Executive.
2. Из меню Утилиты (Utils) запускаем PCAD Symbol Editor.
3. Если требуется создать компонент с неоднородными элементами – на первом этапе создаем символ одного, любого из нужных типов элементов. К остальным вернемся ниже.
4. Загружаем прилагаемый к данной памятке файл SHAB_SYM.sim - шаблон, на основе которого будем создавать свой символ. Шаблон содержит все реально необходимые метрические сетки и текстовые стили с использованием русских шрифтов.
5. Первым делом, командой Save To File As (Сохранить в файл как) из меню Symbol (Символ), создаем рабочую копию шаблона с нужным названием.
6. Далее – рисуем символ и убираем с шаблона все ненужное.

Для изображения выводов используем инструмент Place Pin (Размещение пина), при выборе которого сразу же предлагается следующее меню:

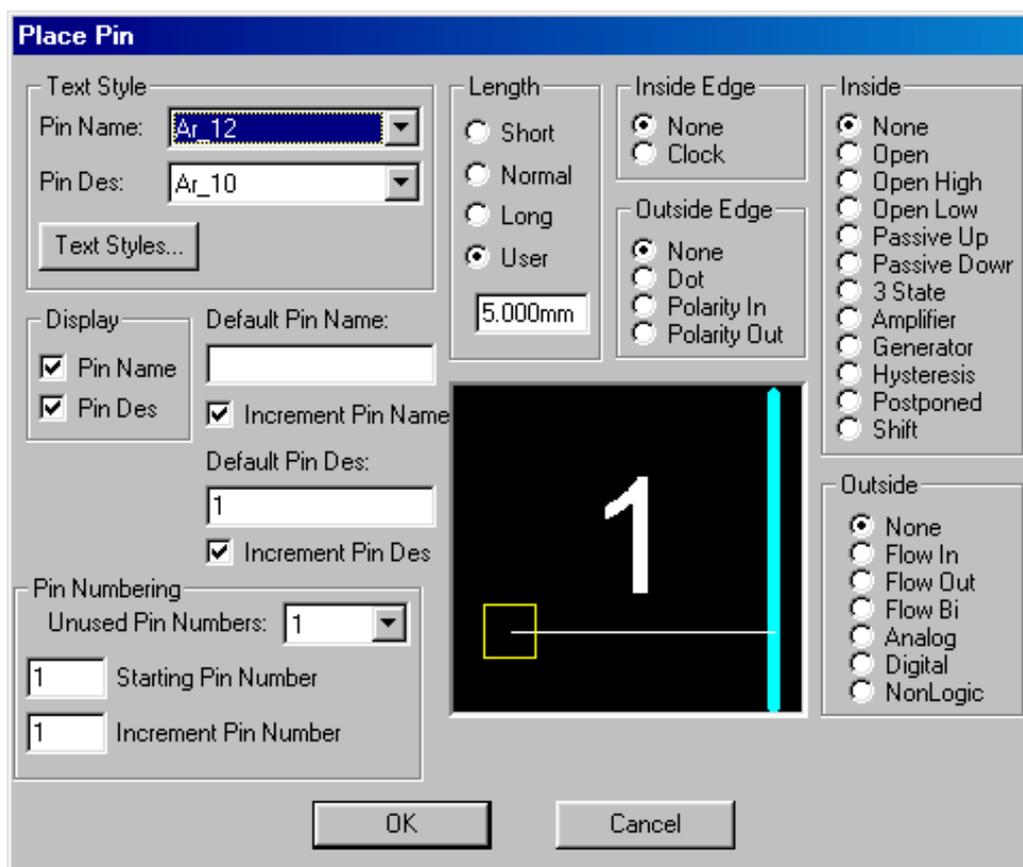


Назначение полей (после некоторого обдумывания) вполне очевидно.

Поля Pin Des (Обозначение вывода) и Pin Number (Номер вывода) описывают в конечном итоге номер вывода, изображаемый вне поля УГО, а поле Pin Name — его функцию, изображаемую внутри УГО.

ВНИМАНИЕ! По умолчанию, буквы, введенные в поле Pin Name, оказываются слишком прижатыми к линии, ограничивающей символ. И при левом размещении вывода и при правом. Хочу предостеречь читателя от соблазна придать более красивый вид символу путем введения дополнительных пробелов. Главное, что эти пробелы без проблем вводятся, отражаются на изображении символа и более того – сохраняются в "первичном" компоненте, создаваемом PCAD Symbol Editor. Но после обработки компонента инструментами PCAD Library Executive эти пробелы (ТОЛЬКО ОНИ!), к сожалению, исчезают! Этот печальный опыт стоил мне полдня нервных экспериментов.

Рекомендую использовать следующие настройки (на примере вывода, являющегося входом аналогового ключа). Изменены поля: Text Style (Стиль текста), Length (Длина), Pin Des (Обозначение). С остальными полями предлагаю поиграть на досуге!

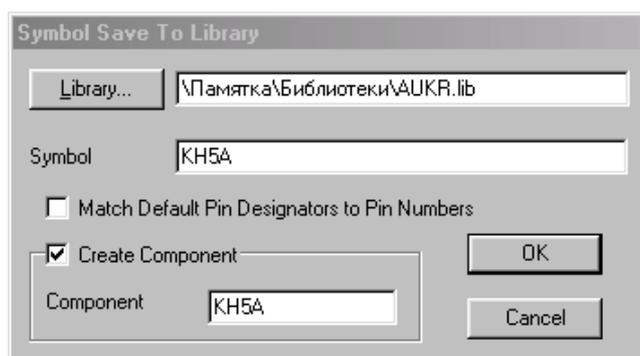


7. По окончании рисования, сохраняем *неупакованный* файл созданного символа использовавшейся ранее командой Save To File As (Сохранить в файл как) из меню Symbol. Это может пригодиться при возникновении в будущем желаний что-то поменять в размерах.

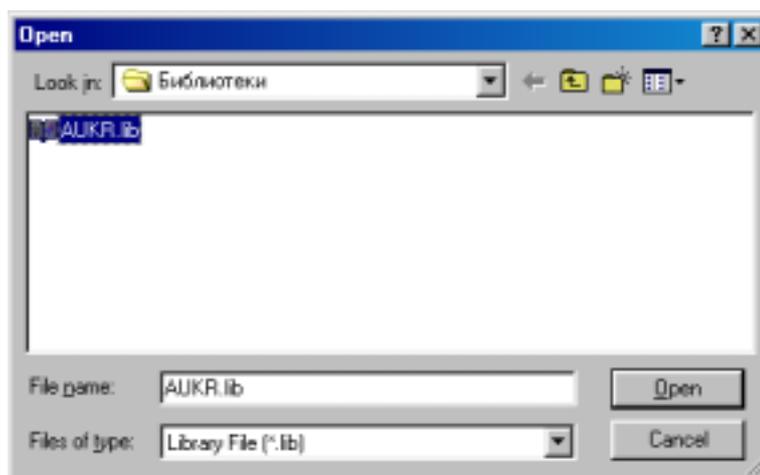
Пример неупакованного символа — прилагаемый файл KH5a.sym.

8. Сохраняем символ *в библиотеку* командой Save As (Сохранить как) из меню Symbol.

При этом раскрывается окно:



Нажатием на кнопку Library вызывается еще одно окно, в котором можно выбрать библиотеку, в которую будет помещен создаваемый символ и компонент на его основе.

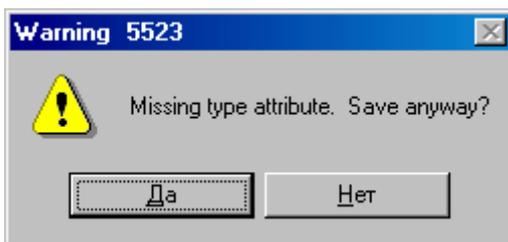


Стандартным для WINDOWS образом находим и открываем нужную библиотеку.

После этого, в предыдущем (по этому тексту) окне, находим нужное поле и задаем имя символу. Если "сброшено" окошко Create Component — отмечаем его "птицей" и в соответствующем поле присваиваем имя компоненту.

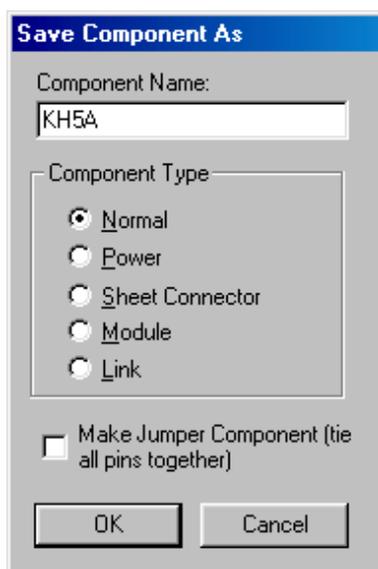
9. Далее, после нажатия OK, редактор предупредит Вас, если *не заданы* все *обязательные* атрибуты. В подобном случае процедура сохранения прервется, придется задать эти атрибуты и начать сохраняться заново.

В том случае, если все обязательные компоненты заданы, но не заданы необязательные атрибуты – появится сообщение:



Нажав кнопку "Да", можно продолжить сохранение без задания типа ЭРЭ.

10. Далее, редактор предложит следующее меню:



За четыре года моего общения с PCAD у меня не возникло необходимости разбираться – что дает этот выбор, чего и Вам желаю.

Поэтому, на данное окно можно пока не обращать внимания и жать на ОК.

11. В результате содеянного, в библиотеке AUKR.lib оказывается:

- символ (в нашем примере - KH5a), который можно вызвать редактором символов;
- компонент (KH5A), который можно вызвать и подкорректировать библиотечным менеджером.

12. Если предполагается создание неоднородного компонента, следует повторить весь путь от п. 4 до п. 10 для элементов, имеющих иное начертание. Пример – символы KH5b и KH5c в неупакованных файлах KH5b.sym и KH5c.sym и в библиотеке.

13. Переходим к созданию конечного продукта – реального компонента.

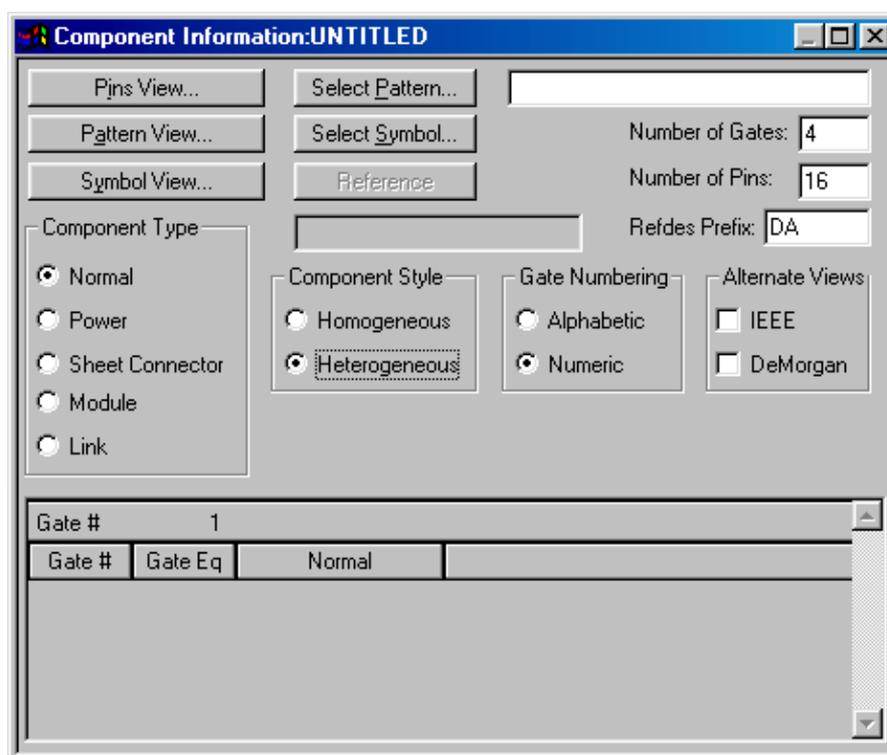
ВНИМАНИЕ! Все дальнейшие действия требуют хирургической точности и аккуратности, поскольку при любой ошибке мне приходилось каждый раз начинать сначала – именно с этого места.

14. Переходим в программу PCAD Library Executive.

15. Для однородного компонента, командой Open (Открыть) из меню Component, открываем созданный ранее "первичный" компонент.

Для неоднородного компонента можно использовать "первичный" компонент одного из входящих символов, но лучше все-таки командой New из меню Component создать новый компонент с другим именем, например – просто КН5.

Известным уже нам способом, программа запросит имя библиотеки, с которой мы хотим работать, а затем предложит свое рабочее окно:



16. А теперь, аккуратненько:

— в поле Number of Gates вносим число элементов, например – 4;

— в поле Number of Pins вносим число выводов, например – 16;

— в поле Refdes Prefix вносим код элемента по ГОСТ 2.710, например – DA;

— отмечаем опцию Numeric в поле Gate Numbering (Нумерация секций);

— в зависимости от того – однородный или нет создаваемый компонент – отмечаем требуемую опцию в поле Component Style (Стиль компонента). В нашем примере — Heterogeneous (Неоднородный);

— в появившейся (по ходу набора) таблице в нижней части окна, редактируем столбец Gate Eq, отражающий использование однотипных элементов.

Для двухэлементного неоднородного компонента, имеющего один элемент одного типа и второй - другого, таблица должна иметь вид:

Gate #	Gate Eq	Normal
1	1	
2	2	

Для однородного четырехэлементного компонента:

Gate #	Gate Eq	Normal
1	1	
2	1	
3	1	
4	1	

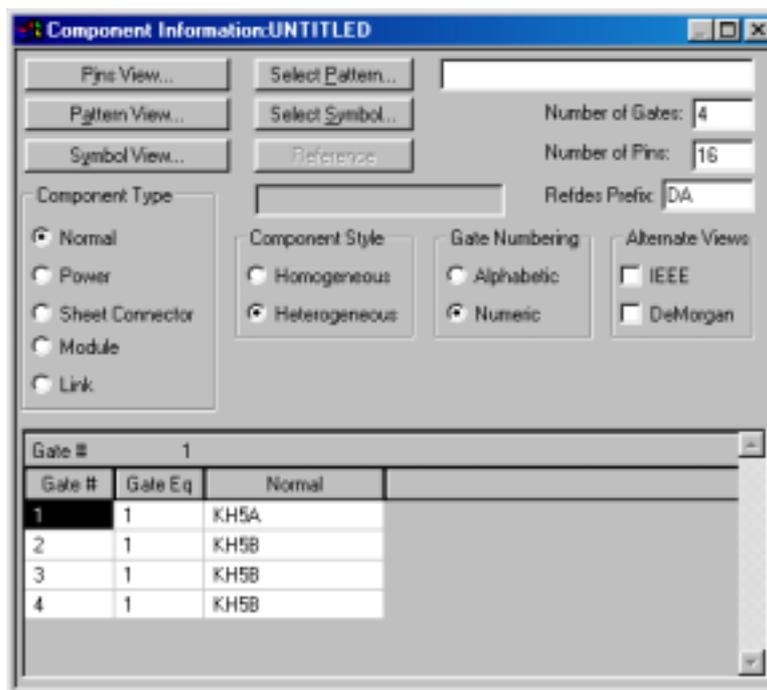
В нашем случае:

Gate #	Gate Eq	Normal
1	1	
2	2	
3	2	
4	2	

16. Поочередно, для каждой строки таблицы, выделяем мышкой ячейку в столбце Normal, а затем нажимаем кнопку Select Symbol (Выбор символа) и выбираем требуемый символ.

В нашем примере, для Gate # 1 выбираем KH5a, а для Gate # 2, 3 и 4 выбираем KH5b.

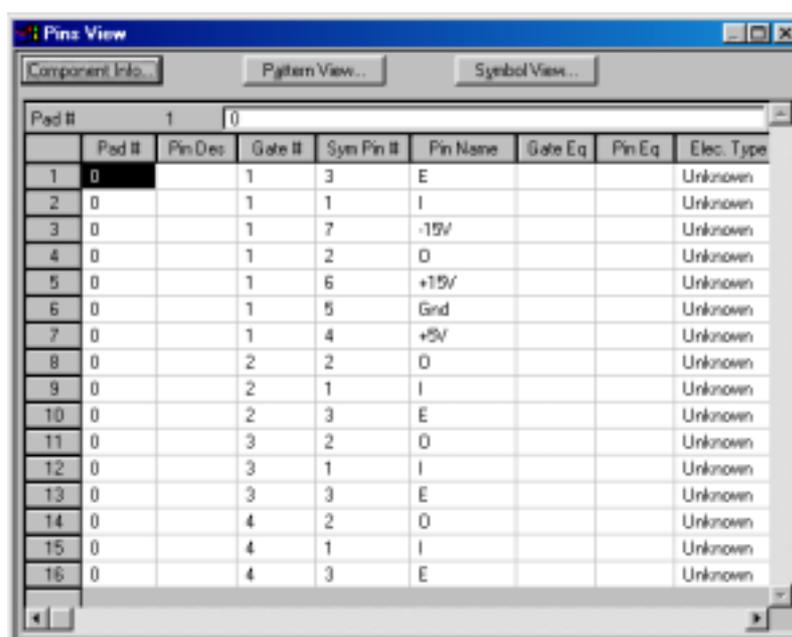
Окно должно приобрести следующий вид.



Примечание. Если Вам к этому времени уже доступен файл топологического посадочного места (pattern) компонента для разводчика печатной платы, его можно подключить к создаваемому компоненту, вызвав соответствующее меню нажатием кнопки Pattern View (Просмотр посадочного места).

17. Нажимаем кнопку Pins View (Просмотр вывода).

Появляется окно, в котором, необходимо поставить в соответствие условные номера выводов символов элементов (Sym Pin #) реальным номерам выводов корпуса компонента (Pad # и Pin Des).



Столбцы Pad# и Pin Des следует заполнить вручную, пользуясь справочными данными, после чего окно приобретает вид:

	Pad #	Pin Des	Gate #	Sym Pin #	Pin Name	Gate Eq	Pin Eq	Elec. Type
1	3	3	1	1	I	1		Unknown
2	2	2	1	2	O	1		Unknown
3	1	1	1	3	E	1		Unknown
4	12	12	1	4	+5V	1		Unknown
5	5	5	1	5	Gnd	1		Unknown
6	13	13	1	6	+15V	1		Unknown
7	4	4	1	7	-15V	1		Unknown
8	6	6	2	1	I	2		Unknown
9	7	7	2	2	O	2		Unknown
10	8	8	2	3	E	2		Unknown
11	14	14	3	1	I	2		Unknown
12	15	15	3	2	O	2		Unknown
13	16	16	3	3	E	2		Unknown
14	11	11	4	1	I	2		Unknown
15	10	10	4	2	O	2		Unknown
16	9	9	4	3	E	2		Unknown

В нашем примере выводы питания включены в символ первого элемента и в "распиновке" компонента учтены автоматически.

При создании компонента с однородными элементами, для описания выводов питания придется дополнительно поработать ручками. Покажем это на примере той же микросхемы 590КН5, для которой создадим компонент КН5_БЕЗ_ПИТАНИЯ (в отличие от КН5), состоящий из четырех однотипных символов КН5в.

Рассмотренные выше окна PCAD Library Executive будут иметь вид:

Gate #	Gate Eq	Normal
1	1	КН5В
2	1	КН5В
3	1	КН5В
4	1	КН5В

а для выводов

	Pad #	Pin Des	Gate #	Syn Pin #	Pin Name	Gate Eq	Pin Eq	Elec. Type
1	3	3	1	2	0			Unknown
2	2	2	1	1	1			Unknown
3	1	1	1	3	E			Unknown
4	6	6	2	2	0			Unknown
5	7	7	2	1	1			Unknown
6	8	8	2	3	E			Unknown
7	14	14	3	2	0			Unknown
8	15	15	3	1	1			Unknown
9	16	16	3	3	E			Unknown
10	11	11	4	2	0			Unknown
11	10	10	4	1	1			Unknown
12	9	9	4	3	E			Unknown
13	12	12	PwR		+5V			Power
14	5	5	PwR		GND			Power
15	13	13	PwR		+15V			Power
16	4	4	PwR		-15V			Power

Четыре последние строки в таблице выводов необходимо заполнить вручную, при этом непосредственно набором заполняются столбцы Pad #, Pin Des и Pin Name.

Процесс набора, подобно электронной таблице Microsoft Excel, отражается в служебной строке, в верхней части таблицы.

Для заполнения столбца Elec. Type, необходимо:

- выделить одну из ячеек;
- нажать на кнопку в правой части служебной строки (с широкой стрелкой вниз);
- из "выпавшего" списка выбрать запись Power и щелкнуть по ней мышкой. Этот текст появится в ячейке столбца Elec. Type, а символы PWR автоматически внесутся в столбец Gate #.

18. Описанием всех выводов питания процесс создания компонента ЗАВЕРШЕН!

Не закрывая окон для описания компонента и его выводов, сохраняем компонент. Делать это лучше командой Save As из меню Component, чтобы контролировать имя создаваемого компонента, а то всякое бывает...

19. В программе PCAD Schematic проверяем внешний вид символа, его позиционное обозначение и т.д.

Пример – файл Проба.sch в прилагаемой к данной памятке папке "Схемы".

20. Для неоднородных компонентов, командой Delete из меню Library в программе PCAD Library Executive, с помощью довольно понятных окон, удаляем из библиотеки промежуточные, "первичные" компоненты, созданные PCAD Symbol Editor, если они не были напрямую использованы при окончательной доводке компонента (см. п.15). Символы не трогаем!