

Intelligent Schematic Input System

Руководство пользователя

Адаптировано к версии 2010 г.

Issue 6.0 - November 2002

© Labcenter Electronics

ICON REFERENCE CHART

Команды печати и работы с файлами



Команды отображения



*) False Origin - фиксированная точка к югу и западу зоны сетки от которой шаг сетки измерен к востоку и северу.

Иконки основного режима



Иконки гаджетов



Двумерная графика



Инструменты разработки

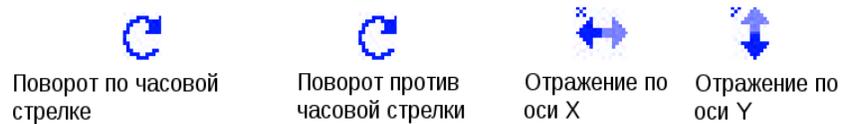


Команды редактирования

Это касается всех отмеченных объектов.



Иконки поворота и отражения



СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ.....	14
О ПРОГРАММЕ ISIS.....	14
ISIS И РАЗВОДКА ПЛАТЫ.....	15
ISIS И СИМУЛЯЦИЯ.....	15
ISIS И СЕТЬ.....	17
КАК ПОЛЬЗОВАТЬСЯ ЭТИМ ДОКУМЕНТОМ.....	17
РУКОВОДСТВО.....	18
ВВЕДЕНИЕ.....	18
ЭКСКУРСИЯ ПО РЕДАКТОРУ ISIS.....	18
ОТБОР, РАЗМЕЩЕНИЕ И СОЕДИНЕНИЕ КОМПОНЕНТОВ.....	20
МАРКИРОВКА И ПЕРЕМЕЩЕНИЕ ОБОЗНАЧЕНИЙ КОМПОНЕНТОВ.....	22
ФУНКЦИИ РЕДАКТИРОВАНИЯ БЛОКА.....	23
ПРАКТИКА — ПУТЬ К СОВЕРШЕНСТВУ.....	23
ОБОЗНАЧЕНИЯ НА ЧЕРТЕЖЕ.....	23
Property Assignment Tool (PAT).....	24
Automatic Annotator.....	25
СОЗДАНИЕ НОВЫХ УСТРОЙСТВ.....	25
ПОСЛЕДНИЕ ШТРИХИ.....	28
СОХРАНЕНИЕ, ПЕЧАТЬ И ВЫВОД НА ПЛОТТЕР.....	29
ПОДРОБНЕЕ О СОЗДАНИИ КОМПОНЕНТОВ.....	30
Создание много-элементных устройств.....	30
Visual Packaging Tool.....	31
Создание похожих устройств.....	33
Замена компонентов в чертеже.....	33
СИМВОЛЫ И БИБЛИОТЕКА СИМВОЛОВ.....	33
СОЗДАНИЕ ОТЧЕТОВ.....	34
БОЛЬШИЕ ПРОЕКТЫ.....	34
ОБЩИЕ ПОНЯТИЯ.....	36
ВИД ЭКРАНА.....	36
Основное меню.....	36
Инструментальные панели.....	36
Инструментальная панель команд.....	36
Панель выбора режима.....	37
Панель ориентации.....	37
Окно редактирования.....	37
Панорамирование.....	37
Увеличение/Уменьшение.....	38
Переменное масштабирование.....	38
Окно обозревателя.....	38
Окно выбранных объектов.....	40
Отображение координат.....	40
СИСТЕМА КООРДИНАТ.....	41

Точка отчета.....	41
Сетка.....	41
Привязка к сетке.....	41
Real Time Snap	41
КОМАНДЫ ХРАНЕНИЯ.....	43
Начиная новый проект.....	43
Загрузка проекта.....	43
Сохранение проекта.....	44
Импорт/Экспорт секции.....	44
Выход из ISIS.....	44
ОБЩИЕ ВОЗМОЖНОСТИ РЕДАКТИРОВАНИЯ.....	44
Размещение объектов.....	44
Чтобы поместить объект:.....	44
Выделение объекта.....	45
Удаление объекта.....	45
Перетаскивание объекта.....	45
Перетаскивание этикеток объекта.....	45
Чтобы переместить этикетку:.....	45
Изменение размеров объекта.....	46
Чтобы изменить размеры объекта:.....	46
Изменение ориентации объекта.....	46
Чтобы изменить ориентацию объекта:.....	46
Редактирование объекта.....	46
Для редактирования единственного объекта с помощью мышки:.....	47
Редактирование объекта с доступом к специальным режимам:.....	47
Для редактирования компонента по имени:.....	47
Редактирование этикеток объектов.....	47
Для редактирования этикетки единичного объекта с помощью мышки:.....	47
Копирование всех выделенных объектов.....	47
Для копирования секции схемы:.....	47
Перемещение выделенных объектов.....	48
Чтобы переместить несколько объектов:.....	48
Удаление всех выделенных объектов.....	48
Чтобы удалить группу объектов:.....	48
ПРОВЕДЕНИЕ СОЕДИНЕНИЙ.....	49
Размещение проводов.....	49
Чтобы провести соединение между двумя объектами:.....	49
Wire Auto-Router.....	49
Повтор соединения.....	49
Перетаскивание проводников.....	50
Чтобы переместить сегмент или группу сегментов соединения:.....	50
Чтобы удалить петлю из проводника:.....	51
АВТОМАТИЧЕСКИЙ АННОТАТОР.....	51
Расстановка значений.....	52
РАЗНОЕ.....	52
Рамка чертежа.....	52
Штамп.....	53

На задний план/На передний план.....	54
ГРАФИЧЕСКИЕ И ТЕКСТОВЫЕ СТИЛИ.....	55
ВВЕДЕНИЕ.....	55
РУКОВОДСТВО.....	56
Редактирование глобальных стилей.....	56
Редактирование локальных стилей.....	58
Работа с шаблонами (Template).....	60
ШАБЛОНЫ И МЕНЮ TEMPLATE.....	61
СВОЙСТВА.....	62
ВВЕДЕНИЕ.....	62
СВОЙСТВА ОБЪЕКТОВ.....	62
Системные свойства.....	62
Пользовательские свойства.....	63
Чтобы редактировать пользовательские свойства объектов:.....	63
Определение свойств (PROPDEFS).....	63
СВОЙСТВА ЛИСТА.....	64
Введение.....	64
Определение свойств листа.....	64
Правила области для свойств листа.....	65
СВОЙСТВА ПРОЕКТА.....	66
Для создания списка свойств проекта:.....	66
ПАРАМЕТРИЗОВАННЫЕ ЦЕПИ.....	67
Введение.....	67
Пример.....	67
Замещение свойств.....	68
Вычисление выражения свойства.....	69
Функции округления E12 (), E24 ().....	70
PROPERTY ASSIGNMENT TOOL.....	70
Диалоговое окно PAT.....	70
Действия PAT.....	71
Режимы работы PAT.....	72
Команда Search and Tag.....	73
Примеры.....	74
Чтобы маркировать ответвления шины:.....	74
Чтобы присвоить корпуса для всех BC108 в проекте:.....	74
Для переименования всех ITEM свойств в свойство CODE:.....	74
Чтобы скрыть все свойства упаковки:.....	74
Чтобы присвоить большие корпуса конденсаторам 1000 мкФ:.....	75
ОПРЕДЕЛЕНИЕ СВОЙСТВ.....	75
Создание Property Definitions (определение свойств).....	75
Предопределенные определения свойств.....	75
Старые проекты.....	76
СПЕЦИФИКАЦИЯ ОБЪЕКТА.....	78
КОМПОНЕНТЫ.....	78
Выбор компонентов из библиотек устройств.....	78
Чтобы выбрать компоненты из библиотек устройств:.....	78
Размещение компонентов.....	79

Чтобы разместить компонент:.....	79
Замещение компонентов.....	79
Для замещения компонента:.....	79
Редактирование компонентов.....	80
Свойства компонентов.....	80
Скрытые выводы питания.....	80
ТОЧКИ.....	80
Размещение точек.....	81
Автоматическое размещение точек.....	81
Автоматическое удаление точек.....	81
МАРКЕРЫ ПРОВОДОВ.....	81
Размещение и редактирование этикеток проводов.....	82
Чтобы разместить или отредактировать этикетку провода:.....	82
Чтобы изменить вид соединения:.....	82
Удаление маркировки проводов.....	83
Чтобы удалить этикетку провода:.....	83
Использование маркировки провода для доступа к имени сети.....	83
Использование этикеток проводов для доступа к свойствам сети.....	83
Свойства этикетки провода.....	84
SCRIPTS.....	84
Размещение и редактирование скриптов.....	84
Чтобы разместить скрипт:.....	84
Чтобы отредактировать скрипт:.....	85
Типы блоков скриптов.....	85
Назначение свойств элемента (*FIELD).....	85
Назначения глобального свойства сети листу (*NETPROP).....	86
Определение свойства листа (*DEFINE).....	86
Таблицы картирования параметров (*MAP ON varname).....	87
Таблицы определения модели (*MODELS).....	87
Именованные скрипты (*SCRIPT scripttype scriptname).....	88
Скрипт модели SPICE (*SPICE).....	89
ШИНЫ.....	89
Размещение шин.....	89
Этикетки шин.....	89
Для удаления этикетки шины:.....	90
Для изменения вида шины:.....	90
Соединение проводов/шин.....	91
Чтобы разместить отвод на шине:.....	91
Чтобы маркировать ответвление шины:.....	91
Свойства шины.....	91
ПОДСХЕМЫ.....	92
Размещение подсхем.....	92
Чтобы разместить прямоугольник подсхемы:.....	92
Чтобы разместить порты подсхемы:.....	92
Редактирование подсхем.....	93
Свойства подсхем.....	93
ВЫВОДЫ.....	93

Логические выводы.....	94
Физические выводы.....	94
Размещение выводов.....	94
Чтобы поместить вывод:.....	95
Редактирование выводов.....	95
Свойства выводов.....	96
КОНТАКТЫ.....	96
Размещение контактов.....	96
Чтобы разместить контакт:.....	97
Редактирование контактов.....	97
Свойства контактов.....	97
2D GRAPHICS (Графические элементы).....	98
Размещение графических элементов.....	98
Чтобы поместить линию:.....	98
Чтобы разместить прямоугольник:.....	98
Чтобы разместить круг:.....	98
Чтобы разместить дугу:.....	99
Чтобы разместить произвольную фигуру (path):.....	99
Чтобы разместить графический текст:.....	100
Чтобы разместить символ:.....	100
Изменение размеров графических объектов.....	100
Редактирование графических объектов.....	101
МАРКЕРЫ.....	101
Типы маркеров.....	101
Размещение маркеров.....	101
Чтобы разместить маркер:.....	102
ВОЗМОЖНОСТИ БИБЛИОТЕКИ.....	104
ОСНОВНОЕ О БИБЛИОТЕКАХ.....	104
Строение библиотеки.....	104
Команда Pick.....	104
БИБЛИОТЕКИ СИМВОЛОВ.....	105
Графические символы.....	105
Чтобы сделать графический символ:.....	105
Определенные пользователем выводы (Terminals).....	106
Чтобы создать определенный пользователем вывод:.....	106
Определенные пользователем порты модуля.....	107
Чтобы создать определенный пользователем порт модуля:.....	107
Определенные пользователем контакты (pins) устройства.....	108
Чтобы сделать определенный пользователем контакт устройства:.....	108
Редактирование существующего символа.....	108
Чтобы редактировать символ:.....	108
Определение иерархических символов.....	109
БИБЛИОТЕКИ УСТРОЙСТВ.....	109
Создание элемента устройство.....	110
Чтобы создать элемент устройства:.....	110
Команда Make Device.....	113
Страница свойств устройства.....	114

Страница упаковки (Packagings).....	114
Страница свойств и определений (Component Properties&Definitions).....	115
Страница справочного листа и подсказки (Data Sheet and Help).....	115
Страница выбора библиотеки.....	116
Visual Packaging Tool.....	117
Packaging Selector (селектор упаковки).....	118
Сетка выводов.....	118
Количество элементов.....	118
Обозреватель упаковки.....	118
Скрытые выводы.....	119
Общие выводы.....	119
Замена вентиляей.....	119
Заменяемые выводы.....	119
Не присоединенные выводы (NC Pins).....	120
Создание одно-элементных устройств.....	120
Чтобы создать устройство с единственным элементом:.....	120
Создание много-элементного гомогенного устройства.....	121
Чтобы сделать много-элементное гомогенное устройство:.....	121
Создание много-элементного гетерогенного устройства.....	121
Чтобы создать много-элементное гетерогенное устройство:.....	121
Пример.....	122
Создание устройства с шинными выводами.....	123
Чтобы создать устройство с шинными выводами:.....	123
Определение свойств и свойства по умолчанию.....	124
Чтобы добавить определенные свойства устройству:.....	124
Для добавления или изменения свойств существующего устройства:.....	124
Распределение выводов питания (Power Pins).....	124
Чтобы создать скрытый вывод на простом единственном элементе:.....	125
Чтобы создать скрытый вывод питания в корпусе:.....	125
Чтобы изменить выводы скрытой сети:.....	126
Редактирование существующих устройств.....	126
Чтобы отредактировать графику или выводы:.....	126
Чтобы отредактировать свойства устройства:.....	126
Чтобы отредактировать упаковку устройства:.....	127
МНОГОСТРАНИЧНЫЙ ПРОЕКТ	128
МНОГОСТРАНИЧНЫЕ ПРОЕКТЫ	128
Введение.....	128
Команды раздела Design.....	128
ИЕРАРХИЧЕСКИЕ ПРОЕКТЫ	128
Введение.....	128
Терминология.....	129
Подсхемы.....	130
Чтобы задать иерархию в подсхеме:.....	130
Модуль-компоненты.....	131
Чтобы установить иерархию модуль-компонента:.....	131
Внешние модули.....	132
Чтобы установить внешний модуль в связке с элементом библиотеки:.....	132

Перемещение по иерархическому проекту.....	132
Глобальные аннотации проекта.....	132
Не физические листы.....	133
ГЕНЕРАЦИЯ NETLIST'a.....	134
ВВЕДЕНИЕ.....	134
ИМЕНА СЕТЕЙ (NET).....	134
ДУБЛИРОВАНИЕ ИМЕН ВЫВОДОВ.....	135
СКРЫТЫЕ ВЫВОДЫ ПИТАНИЯ.....	135
СИНТАКСИС СПЕЦИАЛЬНЫХ ЦЕПЕЙ.....	136
Глобальные сети.....	136
Внутриэлементные соединения для многоэлементных компонентов.....	137
ПРАВИЛА СВЯЗНОСТИ ШИН.....	138
Правило расположения базы.....	138
Использование этикеток шин для изменения правила связности.....	138
Использование контактов шины для внутренних шин.....	139
Соединение с индивидуальными битами.....	140
Ветвление больших шин.....	141
Общие комментарии и предупреждения.....	142
ГЕНЕРАЦИЯ ФАЙЛА NETLIST.....	143
Формат.....	143
Логический/физический/перенос.....	143
Границы.....	143
Глубина.....	143
Ошибки.....	144
ФОРМАТЫ NETLIST'a.....	144
SDF.....	144
BOARDMAKER.....	144
EEDESIGNER.....	144
FUTURENET.....	144
MULTIWIRE.....	145
RACAL.....	145
SPICE.....	145
SPICE-AGE FOR DOS.....	145
TANGO.....	145
VALID.....	146
VUTRAX.....	146
ГЕНЕРАЦИЯ ОТЧЁТОВ.....	147
BILL OF MATERIALS.....	147
Генерация отчёта.....	147
Конфигурация Bill of Materials.....	147
ИМПОРТ ДАННЫХ ASCII.....	148
Команда IF...END.....	149
Команда DATA...END.....	150
ПРОВЕРКА ЭЛЕКТРИЧЕСКИХ СОЕДИНЕНИЙ.....	152
Генерация отчёта.....	152
Сообщения об ошибках ERC.....	152
ГЕНЕРАЦИЯ ПЕЧАТИ.....	155

ВЫВОД НА ПРИНТЕР.....	155
ВЫВОД НА ПЛОТТЕР.....	155
Цвета перьевого плоттера.....	155
БУФЕР ОБМЕНА И ГЕНЕРАЦИЯ ГРАФИЧЕСКИХ ФАЙЛОВ.....	156
Генерация растрового изображения (bitmap)	156
Генерация метафайла.....	156
Генерация файла DXF.....	156
Генерация файла EPS.....	156
ISIS И ARES.....	157
ВВЕДЕНИЕ.....	157
УПАКОВКА (PACKAGING).....	157
Упаковка по умолчанию.....	157
Ручная упаковка.....	158
Автоматическая упаковка.....	158
Использование BOM для помощи с корпусами.....	159
Проверка упаковки (Package Verifier).....	159
Упаковка с ARES.....	160
СВОЙСТВА СЕТЕЙ И СТРАТЕГИИ ТРАССИРОВКИ.....	160
ПРЯМАЯ АННОТАЦИЯ — ИНЖЕНЕРНЫЕ ИЗМЕНЕНИЯ.....	161
Добавление новых компонентов.....	161
Чтобы добавить новые компоненты в проект:.....	161
Удаление существующих компонентов.....	162
Чтобы удалить компоненты из проекта:.....	162
Изменения связности.....	162
Чтобы изменить связность проекта:.....	162
Перемаркировка компонентов и перепакровка вентиляей.....	163
ОБМЕН ВЫВОДОВ/ОБМЕН ВЕНТИЛЕЙ.....	163
Задание обмена выводов и обмена вентиляей для ISIS Library Parts (библиотеки).....	164
Задание обмена выводов для устройства с единственным элементом.....	164
Задание обмена выводов в многоэлементных устройствах.....	164
Чтобы добавить группу обмена выводов в упаковку:.....	165
Задание обмена вентиляей в многоэлементном устройстве.....	165
Ручное выполнение обмена выводами и вентилями в ARES.....	166
Чтобы обменять два вывода или вентиля:.....	166
Оптимизатор обмена вентиляей.....	167
Чтобы использовать оптимизатор обмена вентиляей:.....	167
ПЕРЕМАРКИРОВКА.....	168
Чтобы перемаркировать вручную:.....	168
Чтобы перемаркировать автоматически:.....	168
ВОЗВРАТ ПЕРЕМАРКИРОВКИ В ISIS.....	169
Полуавтоматический возврат перемаркировки.....	169
Полный автоматический возврат перемаркировки.....	169

О ПРОГРАММЕ ISIS

Многие пользователи САД рассматривают ввод схемы, как неизбежное зло в процессе разводки печатной платы, но мы всегда оспаривали эту точку зрения. Сегодня к разводке платы предлагается и автоматическая расстановка деталей, и автоматическая разводка дорожек, что приводит к основным тратам времени за компьютером на ввод схемы. И, если вы используете симуляцию электрических схем для разработки своих идей, то будет лучше потратить больше времени на то, чтобы поработать над схемой.

Программа ISIS была создана именно из этих соображений. Она развивалась на протяжении двенадцати лет исследований и разработки, и была испытана тысячами пользователей во всем мире. Продуманность её архитектуры позволила нам интегрировать первоначальную общепринятую симуляцию, основанную на графике, теперь и с PROTEUS VSM — интерактивной симуляцией электрических цепей в окружении разработки. Появилась возможность нарисовать полную схему на базе микроконтроллера, а затем проверить её работу на той же странице программы в интерактивном режиме. Одновременно, ISIS остаётся основой для приложений разводки печатных плат, так что разработанная схема может экспортироваться в ARES или другое приложение разводки печатных плат.

Для учащихся и инженеров ISIS также подходит в качестве редактора прекрасного вида схем, какие вы видите в журналах. Программа предоставляет полный контроль над видом чертежа в части толщины линий, стилей заливки, цветовой гаммы и шрифтов. Более того, система шаблонов позволяет вам определить «фирменный стиль» и скопировать его из одного чертежа в другой.

Другие общие возможности:

- Работа на Windows 98/Me/2k/XP и более поздних.
- Автоматическое проведение соединений и размещения/удаления точек соединений.
- Мощный инструмент для выбора объектов и назначения им свойств.
- Полная поддержка шин, включая выходы компонентов, точки межлистовых соединений, порты модулей и провода.
- Отчёты о правильности электрических соединений и спецификация материалов (Bill of Materials).
- Спецификация компонентов и соединений (netlist) для применения во всех популярных программах разводки печатных плат.

Для «искушённых пользователей» ISIS включает ряд возможностей, которые помогают в работе над большими проектами. В частности, среди наших покупателей есть те, кто использовал программу при разработке устройств, состоящих из многих тысяч компонентов.

- Иерархическая разработка с поддержкой для компонентов с параметризованными значениями в подсхемах.
- Полная аннотация разработки допускающая множественным образцам подсхем иметь разные номера компонентов.
- Автоматическая аннотация дает возможность нумеровать компоненты автоматически.
- Импорт данных в формате ASCII предоставляет возможность автоматически вносить складские коды компонентов и цены в разработку или файлы библиотеки, где они могут сочетаться или перекрывать данные спецификации.

ISIS И РАЗВОДКА ПЛАТЫ

Пользователи ARES или, в действительности, другой программы разводки печатных плат могут найти несколько интересных для себя возможностей:

- Sheet Global Net Properties (общие свойства соединений листа) позволяющие рационально определить стратегию разводки для всех соединений данного листа (то есть, питание с нужной шириной дорожки POWER).
- Физические точки соединения, которые обеспечат средства получить выводы на разъеме, когда они разбросаны по всему чертежу.
- Поддержку гетерогенных многоэлементных компонентов. Например, реле может иметь три компонента, названных RELAY:A, RELAY:B и RELAY:C. RELAY:A — обмотка, тогда как B и C отдельные контакты. Каждый элемент может размещаться индивидуально там, где он улучшает вид чертежа.
- Поддержка обмена выводов и вентиляей. Включает и возможность задать допустимый обмен в библиотеке компонентов ISIS, и возможность изменений обратной аннотации в схеме.
- Инструмент визуализации корпусов, показывающий посадочные места (PCB footprint) и нумерацию выводов около списка названий выводов для элемента схемы. Это облегчает работу и позволяет избежать ошибок при назначении номеров по названиям выводов. Дополнительно можно создать многоэлементный компонент для единственного элемента схемы.

Тому, как использовать ISIS и ARES вместе, посвящена целая глава.

ISIS И СИМУЛЯЦИЯ

ISIS предоставляет развитое окружение для PROTEUS VSM — нашего революционного, интерактивного на системном уровне симулятора. Он сочетает режим смешанной симуляции

электрических цепей, микроконтроллеров и интерактивных моделей компонентов, позволяя симулировать разработки на основе микроконтроллеров.

ISIS предоставляет в первую очередь способ ввода чертежа, архитектуру для интерактивной симуляции в реальном времени и систему обслуживания исходного и объектного кода, связанного с каждым проектом. Вдобавок на схеме может быть размещено множество графических объектов, допускающих выполнение общепринятых переменных симуляции: времени, частоты и развёртки.

Основные возможности PROTEUS VSM включают:

- Симуляцию в смешанном режиме, основанном на Berkeley SPICE3F5, с расширениями для цифровой симуляции и операций реального смешанного режима.
- Поддержку для интерактивной и основанной на графиках симуляции.
- Модели CPU популярных микроконтроллеров, таких как PIC и 8051 серий.
- Интерактивные периферийные модели, включающие LED и LCD дисплеи, универсальную матричную клавиатуру, терминал RS232 и целую библиотеку переключателей, потенциометров, ламп, светодиодов и т. д.
- Виртуальные инструменты, включающие вольтметры, амперметры, двухлучевой осциллограф и 24 канальный логический анализатор.
- Экранные графики — графики размещаются непосредственно на схеме подобно другим объектам. Графики могут быть увеличены на весь экран для измерений с помощью курсора и так выведены.
- Типы анализа, основанного на графиках, включают: переходные процессы, частотный анализ, анализ шума и искажений, развёртку на переменном и постоянном токе и преобразования Фурье. Аудио график позволяет проигрывать симулируемые сигналы.
- Есть прямая поддержка моделей аналоговых компонентов в SPICE формате.
- Открытая архитектура для моделей «plug in» компонентов, кодированных на C++ или других языках. Это может быть электрическая, графическая или модель, комбинирующая оба варианта.
- Цифровой симулятор включает Basic-подобный язык программирования для моделирования и тестирования в технике «vector generation».
- Чертёж, созданный для симуляции, может использоваться и для создания спецификации (netlist) к разводке печатной платы — нет нужды повторно вводить схему.

Все детали, описывающие эти возможности и многое другое, можно найти в руководстве PROTEUS VSM.

ISIS И СЕТЬ

ISIS полностью совмещена с сетью и предлагает следующие возможности для поддержки Network Manager (программы поддержки сети):

- Библиотечные файлы могут быть помечены как «Только для чтения». Это предохранит их от потери символов или устройств, которые могут использоваться другими.
- Есть индивидуальная пользовательская конфигурация ISIS в реестре Windows. Как только реестр определяет локализацию библиотечных файлов, он определяет, что пользователи могут иметь индивидуальные USERDVC.LIB файлы в их персональных или групповых директориях.

КАК ПОЛЬЗОВАТЬСЯ ЭТИМ ДОКУМЕНТОМ

Это руководство предназначено дополнить информацию предоставляемую в режиме онлайн помощи. Несмотря на то, что руководство содержит необходимую информацию, HELP предоставляет контекстную информацию, относящуюся к выбранным иконкам, командам и диалоговым формам. Помощь для большинства объектов в интерфейсе пользователя может быть получена выделением объекта мышкой и нажатием на клавишу **F1** клавиатуры.

ISIS — это большая и чрезвычайно мощная часть программного обеспечения, и специалисту совсем не обязательно использовать все разом. Однако основы того, как ввести схему и создать собственные компоненты, крайне просты, а техника для выполнения этих задач может быть освоена достаточно быстро с помощью руководства, изложенного ниже. Мы очень рекомендуем, чтобы вы поработали над этим, что в дальнейшем сохранит ваше время.

В некоторых расширенных аспектах пакета вы, возможно, найдёте для себя новые концепции, не считая того, как ISIS поддерживает их. Каждой области программного обеспечения присущи свои особенности, и мы, в общем, начинаем с объяснения основ теории, прежде чем перейти к операциям и использованию соответствующих возможностей. Надеемся, вы найдёте заслуживающим внимания эти вводные разделы, вместо того, чтобы сразу «брать быка за рога».

При переводе документа сделаны некоторые добавления и изменения, соответственно более поздним версиям программы.

ВВЕДЕНИЕ

Назначение этого руководства провести вас через процесс ввода не слишком сложной схемы, чтобы дать вам возможность освоиться с техникой управления ISIS. Руководство начинается с самых лёгких задач, как размещение и соединение компонентов, их перемещения, чтобы перейти к таким более тонким операциям, как создание новых элементов библиотеки. Для тех же, кто спешит увидеть что-то «живьём», есть файл ISISTUT.DSN, содержащий полностью укомплектованную учебную схему. Этот и другие проекты установлены в директории SAMPLES.

ЭКСКУРСИЯ ПО РЕДАКТОРУ ISIS

В этом месте мы подразумеваем, что вы установили пакет программ и что текущая директория вполне подходящее место на вашем жёстком диске.

Чтобы начать работать с программой ISIS щёлкните по кнопке «Пуск», выберите раздел «Программы», Proteus 6 Professional (сегодня более поздние версии), а затем ISIS 6 Professional. Редактор схем ISIS загрузится и запустится. В верхней части экрана есть основное меню.

Наибольшая часть экрана отведена под *окно редактора*, и она работает подобно полю для черчения, куда вы помещаете и где соединяете все компоненты схемы. Меньшая область в верхней части справа (в последних версиях слева) называется *окно обзора*. Оно отображает, как следует из его названия, весь чертёж — синий контур показывает края текущего листа, а зелёный область листа в настоящий момент отображаемую в окне редактора. Однако, когда новый объект выбирается из перечня объектов, *окно обзора* показывает вид выбранного объекта. Но об этом позже. Вы можете «подстроить» область черчения в окне редактора несколькими путями:

- Для простого панорамирования *окна редактора* вверх, вниз, влево или вправо установите курсор мышки на выбранную область окна редактирования и нажмите клавишу **F5** на клавиатуре.
- Нажмите и удерживайте клавишу **SHIFT** и толкните мышкой край окна редактирования для панорамирования вверх, вниз, влево или вправо. Мы назвали это *Shift Pan*.
- Если вы хотите переместить *окно редактирования* в совершенно другую часть чертежа, то самый быстрый способ — указать на центр новой области в *окне обзора* и щёлкнуть левой клавишей мышки (в последней версии дважды).
- Вы также можете использовать иконку **Pan** инструментальной панели.

Чтобы настроить масштаб чертежа в *окне редактирования*, вы можете:

- Указать мышкой место, где вы хотите увеличить или уменьшить вид и нажать клавиши **F6** или **F7**, соответственно.

- Нажать клавишу **F8**, чтобы отобразить весь чертеж.
- Удерживая клавишу **SHIFT**, обвести мышкой нужную область чертежа, которая увеличится после отпускания клавиши мышки. Мы назвали это *Shift Zoom*.
- Используйте иконки **Zoom In**, **Zoom Out**, **Zoom Full** или **Zoom Area** инструментальной панели.

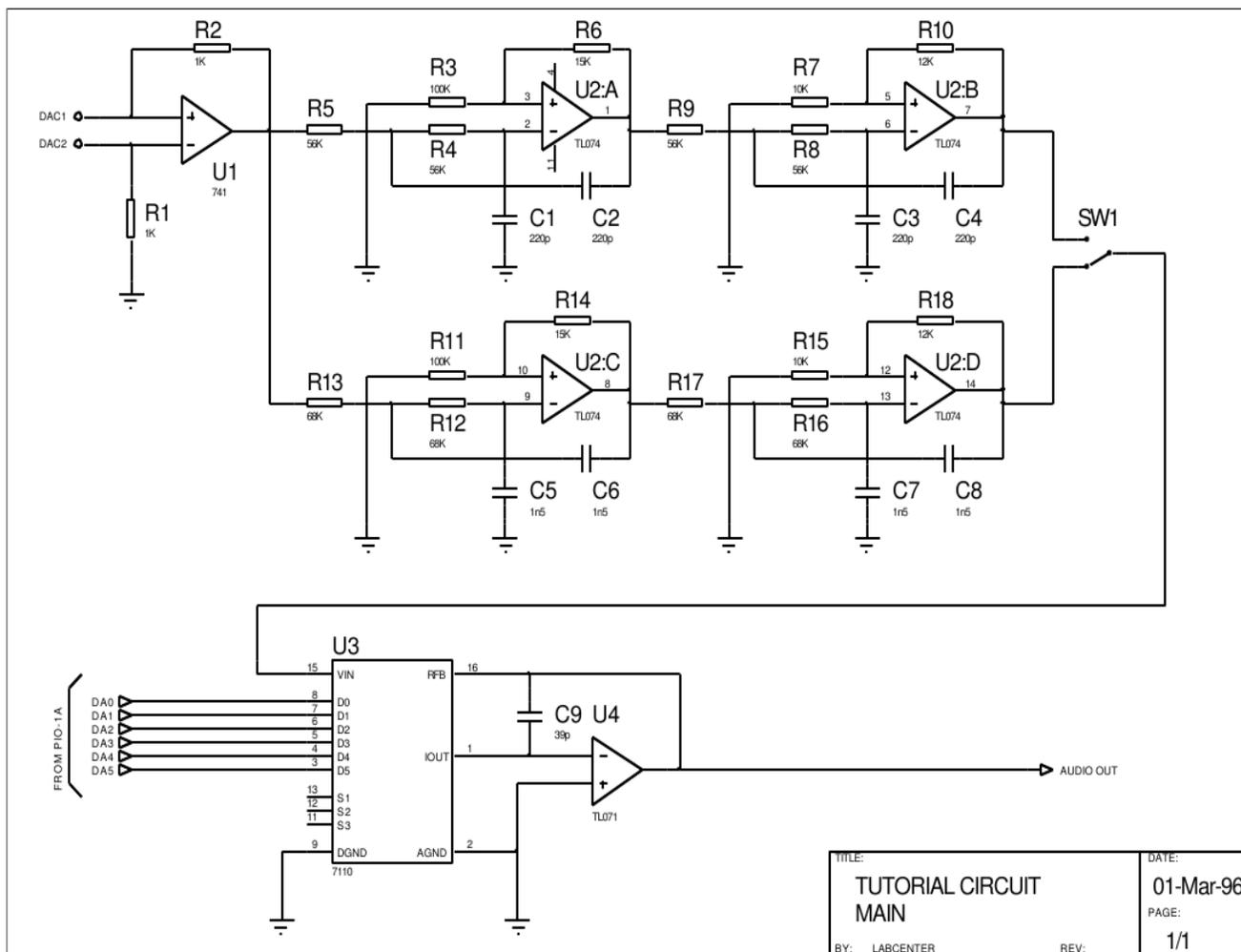
Сетка может быть отображена в *окне редактирования* при использовании команды *Grid* из меню *View*, или после нажатия «**G**», или щелчком по иконке **Grid** инструментальной панели. Сетка помогает выравнивать компоненты и соединения и пугает меньше, чем чистый экран. Если вам не нравится такой вид сетки, вы можете слегка уменьшить контрастность вашего монитора (по умолчанию сетка серая) или изменить цвет с помощью раздела *Set Design Defaults* в *Template* основного меню.

Ниже *окна обзора* находится *навигатор выбранных объектов*, который вы используете для выбора устройств, символов и других библиотечных объектов.

В нижней части справа есть дисплей, отображающий координаты, который считывает соответствующие координаты курсора мышки. Эти координаты в 1 thou units (**1 thou = SI units. 25.4×10^{-6} m. 25.40 μ m**) от центра чертежа.

ОТБОР, РАЗМЕЩЕНИЕ И СОЕДИНЕНИЕ КОМПОНЕНТОВ

Схема, которую мы собираемся нарисовать, показана ниже. Может показаться, что предстоит сделать очень много, но многие элементы схожи (прецизионный на четырех операционных усилителях фильтр), что позволит использовать копирование блоков.



The Tutorial Circuit

Мы начнём с буферного усилителя 741, содержащего U1, R1, R2. Сначала укажем на кнопку **P** в верхней части *навигатора выбора объектов* и щёлкнем левой клавишей мышки. Это приведёт к появлению диалогового *окна выбора устройств* из библиотеки. Теперь вы можете выбрать устройства из множества тех, что есть в библиотеке. А там есть и *объекты*, и *библиотеки*, и *расширения* и *обозреватель*, и не все вы сразу увидите:

- Выбор библиотеки показывает текущее содержание (то есть, DEVICE, TTL, CMOS).
- Выбор объектов отображает все части в текущей библиотеке соответственно с

LABCENTER ELECTRONICS

установками в Extensions (расширения), если это показано (см. ниже).

- Щёлкните левой клавишей мышки по элементу для его отображения или дважды для вставки в чертёж.
- Обзорщик отображает все последние выбранные элементы, как средство просмотра содержания библиотеки.

Вначале нам нужно два элемента: 741 операционный усилитель и резистор (резисторы) для обратной связи. Первый можно найти в разделе *Operational Amplifiers*, второй в разделе *Resistors*. Или можно ввести 741 в окно поиска (*Keywords:*) и просмотреть все варианты, выбрав нужный.

Когда вы выбрали компонент и использовали иконки *поворота* или *отражения* для придания нужной ориентации устройству перед его размещением, его вид, как он будет помещён на схему, отображается в *окне обзора*. Каждый раз при щелчке по иконке поворота влево или вправо, или отражения устройство перерисовывается для предварительного просмотра в новой ориентации. Предварительный просмотр сохраняется до размещения устройства, или новой команды, или действия.

Убедитесь, что операционный усилитель выбран, затем переместите курсор мышки в середину *окна редактирования*. Щёлкните левой клавишей мышки. Появится контур операционного усилителя, который вы можете перемещать с помощью мышки. Последующий щелчок поместит компонент в выбранном месте. Поместите ОУ в центре *окна редактора*. Теперь выберите резистор и поместите его над ОУ. Выберите второй резистор, используя иконку поворота, и поместите его (вертикальный резистор R1), как и предыдущий.

Пока вы не наберётесь опыта, едва ли вы сможете ориентировать и разместить компоненты должным образом, поэтому мы рассмотрим, как перемещать компоненты. В ISIS объекты выбираются для последующего редактирования «тегированием, отметкой». Поместите курсор мышки на операционный усилитель и щёлкните левой клавишей, компонент будет подсвечен. Теперь, нажав левую клавишу мышки, подцепите объект и переместите его, удерживая клавишу. Отпустите её в нужном месте. Это один из способов перемещения компонентов. Щёлкните правой клавишей мышки по ОУ. Второй щелчок правой клавишей мышки удалит объект. Используйте *Undo* в разделе *Edit* основного меню, чтобы вернуть объект. Щёлкните правой клавишей мышки по объекту: из выпадающего меню вы можете выбрать все операции вращения (по часовой и против часовой стрелки, на 180 градусов) и отражения. Вооружившись этими знаниями, вы теперь должны быть в состоянии поместить три компонента, что у вас есть, так, чтобы они соответствовали рисунку выше. Когда вы закончите редактирование, щёлкните левой клавишей мышки где-нибудь в свободной части *окна редактирования*, чтобы снять все выделения объектов.

Теперь мы готовы провести соединения. Начнем с того, что курсор мышки поместим на верхний конец R1 и щёлкнем левой клавишей мышки. ISIS «чувствует», что вы указываете на вывод компонента, и «делает вывод», что вы хотите начать здесь соединение. Чтобы обозначить это, он превращает курсор в зелёный карандаш. Теперь переместите карандаш (он стал серым) к инверсному входу операционного усилителя, где карандаш вновь стал зелёным, и щёлкните там. ISIS определяет это, как завершение соединения. Прделайте это и для R2, следуя рисунку

выше. Попробуйте выделить объекты и поперемещать их, наблюдая, как работает автосоединение (*Wire Auto Router*).

Если вам не нравится автосоединение, вы можете отредактировать его вручную. Чтобы сделать это, выделите провод правым щелчком мышки, а затем переместить, сначала за угол, а затем за середину провода. И, если вы предпочитаете делать соединения вручную, вы можете отключить автосоединение с помощью иконки на инструментальной панели *Toggle Wire Autorouter* (подсказки появляются при наведении курсора на иконку) или командой *Wire Auto Router* в разделе *Tools* основного меню. После этого вы можете почувствовать, когда WAR ведёт себя разумно, а когда вам надо брать все в свои руки.

Для завершения первой части чертежа вам нужно поместить два порта и землю и соединить их, как на рисунке. Чтобы сделать это, выберите иконку *Terminal* (если она есть) или щёлкните правой клавишей мышки и выберите из выпадающего меню *Place->Terminal->GROUND*. Убедитесь, что элемент ориентирован правильно, и поместите его прямо под R1. И там же найдите INPUT и поместите два элемента, как показано на рисунке. В завершение соедините землю с нижним концом R1, а входные порты со входами операционного усилителя. ISIS автоматически создаст требуемые точки соединения, обнаружив, что три провода встречаются в этих точках.

МАРКИРОВКА И ПЕРЕМЕЩЕНИЕ ОБОЗНАЧЕНИЙ КОМПОНЕНТОВ

ISIS имеет очень мощное средство, названное *Real Time Annotation* (оперативное обозначение), которое можно найти в разделе *Tools* основного меню и которое включено по умолчанию. Полную информацию можно найти далее, но, в общих чертах, когда оно включено, компоненты обозначаются при размещении их в схеме. Если вы увеличите любой из резисторов, которые вы поместили в рабочее поле, вы увидите, что ISIS маркирует их и значением по умолчанию, и уникальной ссылкой. Чтобы отредактировать/ввести ссылку компонента и значение щёлкните правой клавишей мышки по нему и выберите из выпадающего меню либо *Edit Label*, если вы щёлкнули по обозначению компонента, либо *Edit Properties*, если щёлкнули по самому элементу схемы. Прделайте это для резисторов R1, 1k и R2, 1k, соответственно. А теперь то же с операционным усилителем и портами. Чтобы переместить этикетки «U1» и «741», как это сделано на рисунке, уменьшите шаг сетки до 50и, нажав на клавишу F2, щёлкните по обозначению, например, «U1», подцепите (наведите курсор на него, нажмите и не отпускайте левую клавишу мышки) этот элемент и перенесите его на новое место в нужную позицию под операционным усилителем. Затем проделайте это с этикеткой «741».

Когда вы закончите позиционирование этикеток, верните шаг сетки к 100, нажав на клавишу F3. Хотя с помощью *Real Time Snap* программа способна найти выводы и провода не с текущим шагом сетки, работа сообразно с тем же шагом сетки сохранит вид чертежа чётким и аккуратным.

ФУНКЦИИ РЕДАКТИРОВАНИЯ БЛОКА

Вы могли заметить, что часть схемы, которую вы нарисовали, в настоящий момент находится в середине чертежа, хотя должна быть в левой верхней части. Чтобы передвинуть её, вначале выделите все объекты, обведя их с помощью мышки: нажмите и удержите левую клавишу мышки, когда курсор находится на свободном месте рабочего поля, и проведите диагональ так, чтобы выделение захватило все объекты. Затем «подцепите» выделение мышкой: выделение меняет цвет, а когда курсор мышки оказывается над выделенной областью, вид курсора меняется, — удерживая левую клавишу мышки, переместите блок в нужное место. Для завершения операции щёлкните левой клавишей мышки на свободном месте чертежа.

Вы можете также заметить, что при перемещении ISIS автоматически панорамирует рабочую область редактора, когда перемещаемый блок выходит за края чертежа.

Пока блок объектов остаётся выделен, вы можете поэкспериментировать с операциями *Block Copy*, *Block Move*, *Block Copy*, *Block Rotate* и *Block Delete* (иконки на инструментальной панели), которые активизируются только при выделении объектов. Отменить операции можно либо нажатием на клавиши **Ctrl+Z**, либо командой (основного меню, раздел *Edit*) *Undo*, либо иконкой инструментальной панели.

ПРАКТИКА — ПУТЬ К СОВЕРШЕНСТВУ

Теперь вы должны приобрести некоторые навыки, так что немного практики в черчении следующей части схемы, сконцентрированной вокруг ОУ U2:A, не помешает. Вам понадобится конденсатор (capasitor). Быстрее всего доступ к выбору устройств, имена которых вы знаете, можно получить с помощью команды *Pick Device/Symbol* — клавиша **P** на клавиатуре, или иконка с буквой P под *окошком обзора*. Используйте разные приёмы редактирования, о которых говорилось выше, чтобы разместить все на своих местах. Передвиньте поля обозначения и значения компонентов в нужное место, но не задавайте обозначений — мы собираемся использовать для этого автообозначение (Automatic Annotator).

Когда вы сделаете один фильтр на ОУ к своему удовлетворению, используйте выделение и иконку *Block Copy*, чтобы сделать три копии — четыре фильтра в одном — как на рисунке. Может оказаться полезным использовать команды масштабирования вида из меню *View* (или горячие клавиши F6 и F7, или иконки инструментальной панели), что даёт возможность видеть весь лист. Когда вы разместите все четыре фильтра, соедините их вместе и добавьте SW-SPDT (достаточно в окне *Keywords*: диалога *Pick Device* ввести искомое), SW1, в схему.

ОБОЗНАЧЕНИЯ НА ЧЕРТЕЖЕ

ISIS предоставляет вам четыре способа для обозначения компонентов:

- Ручное аннотирование — этим методом вы уже пользовались для маркировки первого ОУ и резисторов. Любой объект может редактироваться либо выбором *Edit Properties* в выпадающем меню при щелчке правой клавишей мышки по объекту, либо выделением нужного обозначения правой клавишей мышки и затем *Edit Label*. При появлении

- диалогового окна вы можете в нем изменить подходящим образом Reference (обозначение), Value (величина) и т. д.
- Property Assignment Tool — механизм генерации фиксированного или увеличивающегося значения, текст которого будет закреплён либо за всеми объектами, либо за всеми выделенными объектами (на всех или на текущем листе), либо за объектами, по которым вы потом щёлкаете левой клавишей мышки. Использование PAT ускоряет аннотирование, оно быстрее ручного, но медленнее автоматического. Однако сохраняет за вами контроль, какие имена получают какие элементы.
- Automatic Annotator — использование автоматического обозначения гарантирует, что весь чертёж будет аннотирован за секунды. Инструмент учитывает много-элементные компоненты, как вентиль 7400 TTL NAND, и разумным образом локализует каждую часть. Однако весь процесс не интерактивный, так что вам придётся брать на себя последующий контроль за названиями, используя другие два метода.
- Real Time Annotation — этот способ, когда разрешён, аннотирует компоненты при размещении их на чертеже, избавляя вас от необходимости вводить обозначение и номинал при разработке. Но, как и в случае с автоматической аннотацией, процесс не интерактивный, и не поддаётся вмешательству со стороны пользователя. *Real Time Annotation* может быть включено или выключено командой основного меню в разделе *Tools* или горячими клавишами **CTRL+N**.

На практике вы можете смешивать все четыре метода в любой последовательности. Automatic Annotator может устанавливаться в режим, когда не затрагиваются существующие аннотации, так что вы фиксируете обозначения некоторых элементов, предоставляя ISIS разобраться с остальными. Поскольку Real Time Annotation включено по умолчанию, мы оставляем механизм включённым и используем другие три метода для редактирования имеющихся обозначений на чертеже.

Property Assignment Tool (PAT)

Давайте предположим, что вы хотели бы предварительно обозначить все резисторы, используя PAT. Положим, у вас уже вручную обозначены R1 и R2, и вам нужно генерировать последовательно R3, R4, R5 и т. д. Чтобы сделать это, выберите опцию Property Assignment Tool на инструментальной панели или в основном меню. Введите REF=R# в поле String:, затем переместите курсор мышки в следующее поле Count: и введите значение 3. Убедитесь, что флажок *On Click* установлен и нажмите кнопку **OK** или клавишу **ENTER**. Значок номера «#» в текстовом поле String: будет заменён текущим значением из поля Count: каждый раз, когда PAT даёт значение объекту, а значение поля Count: после этого увеличивается.

ISIS автоматически выбирает нужный режим, так что вы можете обозначать требуемые объекты щелчком слева от них. Переместитесь к резистору R3 и щёлкните слева. PAT выдаст текст R3 и объект перерисует. А теперь повторите это с резистором ниже, R4, и посмотрите, как увеличивается поле Count: каждый раз, когда вы используете PAT. Теперь вы можете аннотировать все оставшиеся резисторы на тот же манер.

LABCENTER ELECTRONICS

Когда закончите с этим, остановите *PAT*: вызовите его диалог и либо щёлкните по кнопке **CANCEL**, либо нажмите клавишу *ESC*.

PAT может также использоваться для назначения тех же String для нескольких выделенных объектов, например, величины резисторов или конденсаторов, которые имеют одинаковое значение. Это относится к конденсаторам C1 и C2, которые имеют одинаковое значение 220 пФ. Чтобы задать эту величину, вначале убедитесь, что только эти конденсаторы выделены: щёлкните на свободном месте чертежа, снимая все выделения, а затем выделите оба конденсатора. Теперь вызовите *PAT* и введите VAL=220p в поле String:, выберите *Local Tagged* опцию (установите флажок в разделе *Apply To:*) и щёлкните **OK**. Это все — вам не нужно останавливать работу *PAT*, поскольку режим маркировки *On Click* выключен.

Попробуйте сами проделать это для остальной части чертежа, пока не уясните, как работает *PAT* — хотя вначале это может показаться мудрёным, в конечном счёте *PAT* оказывается мощным инструментом, который избавит вас от значительной части утомительной работы при редактировании. И не забудьте, что при использовании режима *On Click* вам по окончании работы следует выключить режим.

Automatic Annotator

Механизм Automatic Annotator обеспечивает обозначение компонентов. Он может аннотировать все компоненты или только те, что не были обозначены, то есть, те, что имеют «?» в ссылке.

Если вы уже обозначили некоторые элементы, мы запустим Automatic Annotator в режиме Incremental (последовательное увеличение). Чтобы это сделать, используйте команду *Global Annotator* из раздела *Tools* основного меню, установите флажок *Incremental* и щёлкните по **OK**. Через некоторое время чертёж будет перерисован с новыми обозначениями. Поскольку операционные усилители не много-элементные, как TL074, аннотатор обозначает их от U2 до U5, что не совсем то, что нам нужно. Чтобы это поправить, отредактируем их по очереди в нужных обозначениях. Посмотрим, как создать и использовать присущее TL074 впоследствии.

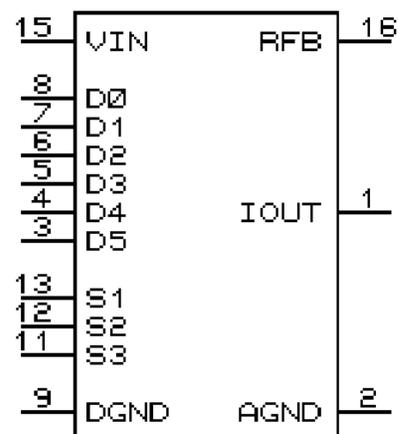
Даже, используя автоматический аннотатор, вы все ещё должны вручную задать значения, но попробуйте ускорить это — вместо возни с каждым компонентом чертежа, нажмите *Find and Edit Component* в разделе *Edit* основного меню и введите обозначение компонента. Это автоматически локализует нужный элемент и откроет его диалоговое окно редактирования. Вы также можете попробовать использование Property Assignment Tool, как описано выше.

СОЗДАНИЕ НОВЫХ УСТРОЙСТВ

В следующей части схемы применяется цифровой аттенюатор 7110, что даст нам возможность познакомиться с тем, как в ISIS создаются новые устройства.

В ISIS новые устройства создаются прямо на чертеже — нет режима отдельного редактора устройств, не говоря об отдельной программе. Новое устройство создаётся с помощью графических средств, выводов, обозначения этих выводов и, наконец, выделением всего и вызовом команды *Make Device* (иконкой на инструментальной панели).

Вы найдёте весьма полезным, когда при создании новых устройств вы видите, как оно выглядело бы на бумаге, и можете установить его примерные размеры, чтобы удобно разместить выводы с каждой стороны и т. д. В этом случае вы можете использовать приведенную диаграмму в качестве образца. Первое, что нам нужно сделать, это найти свободное место в рабочей области, где будет создано новое устройство. Щёлкните левой клавишей мышки в левой нижней области *окна обзора*, чтобы перейти в редакторе в это место.



Начнём с отрисовки основного контура нового устройства.

Выберите иконку *Box* на левой инструментальной панели. Обратите внимание, *окно обозревателя* приобретает графический стиль. Графический стиль определяет, как графика, которую мы собираемся рисовать, будет выглядеть в плане цвета линии, толщины линии, стиля заливки, цвета заливки и т. д. Каждый стиль в списке — это разный набор таких атрибутов и определяет вид будущих разных элементов схемы.

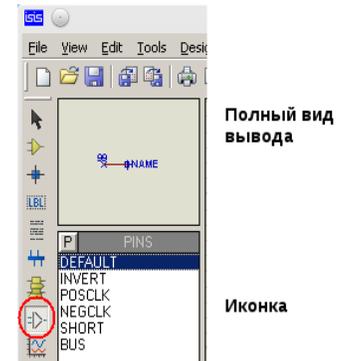
ISIS поддерживает мощную графическую систему стилей, как локальных, так и глобальных; и возможность для локальных стилей «следовать» или «направляться» глобальными стилями, что позволяет вам легко и гибко управлять видом вашей схемы. Посмотрите раздел «Графические и текстовые стили» далее для полного понимания того, как работают стили, и как их использовать.

Поскольку мы рисуем основу компонента, выберите графический стиль *COMPONENT* и затем переместите указатель мышки в окно редактора, щёлкните левой клавишей мышки и протащите указатель по диагонали до получения прямоугольника; щёлкните ещё раз. Не беспокойтесь о «правильных» размерах прямоугольника — вы можете всегда изменить их позже. Посмотрите, результатом выбора стиля *COMPONENT* будет то, что прямоугольник имеет тот же цвет, заливку и т. п., что и другие компоненты схемы.

Следующее, что нужно сделать — это разместить выводы для нового устройства. Чтобы это сделать, вначале выберите иконку *Device Pin* (например, из выпадающего меню после щелчка правой клавишей мышки в разделе *Place*). Справа появится список всех типов доступных выводов (и вы можете создать свой собственный тип в ISIS, хотя мы этого не описываем в данном руководстве). Выберите *Default* тип выводов.

LABCENTER ELECTRONICS

Окно обзора показывает вид вывода с именем и номером, представленными строками NAME и 99, базой и концом, обозначенными маркером и перекрестием, соответственно, при этом перекрестие представляет конец вывода, к которому вы в последствии будете «крепить» провод соединения. Используйте иконки поворота и отражения для ориентации вывода, а затем щёлкните мышкой в рабочем поле и ещё раз в том месте, где выводу надлежит быть. Разместите выводы для VIN, D0..D5, S1..3 и DGND. Заметьте, что можно использовать курсорные клавиши клавиатуры для перемещения на одну клетку вверх, вниз, влево и вправо, а клавишу **ENTER**, чтобы зафиксировать положение; так может получаться быстрее, чем при использовании мышки.



Теперь используйте иконку *Mirror* для получения выводов, расположенных справа: RFB, IOUТ и AGND. Чтобы завершить работу, поместите два вывода, сверху и снизу прямоугольника, используя иконки поворота и отражения; эти выводы предназначены для подключения питания: VDD и VBB, они в конечном счёте должны быть скрытыми (и по этой причине они не показаны на рисунке).

На этой стадии вы можете перепозиционировать выводы или изменить размеры, как вам нужно, прямоугольника. Чтобы переместить вывод, пометьте его нажатием правой клавиши мышки и выберите из выпадающего меню *Drag Object*, завершите все щелчком левой клавиши мышки; для переориентации, используйте иконки поворота и отражения. Для изменения размера прямоугольника выделите его, поместив курсор к его краю; переместите курсор мышки на квадратики, появившиеся по контуру прямоугольника, при этом курсор меняет вид, показывая направление, в котором будет происходить изменение; подцепите (нажмите и удержите) левой клавишей мышки и измените размер до нужного. Если вы изменили ширину, вам понадобится выделить все выводы и переместить их все вместе на новое место.

Итак, прямоугольник основания устройства подправлен, выводы на месте, и пришло время обозначить выводы, добавив названия и номера, и задать им электрический тип. Электрический тип вывода (вход, питание, подтяжка и т. д.) используется инструментом проверки электрических соединений (*Electrical Rules Check*), чтобы гарантировать, что соединены только выводы правильного электрического типа.

Вначале мы зададим имена, электрический тип и видимость. Чтобы это сделать, мы должны выделить каждый вывод, щёлкнув по нему, например, правой клавишей мышки и выбрав из выпадающего меню *Edit Properties* или открыть диалоговое окно свойств вывода двойным щелчком по выводу.

Отредактируйте каждый вывод следующим образом:

- Введите название вывода в поле *Pin Name*. Оставьте поле *Default Pin Number* пустым, поскольку мы назначим номера выводам с помощью механизма *Property Assignment Tool*.

- Выберите подходящий электрический тип вывода: Output для выходов, Power для VDD, VBB, AGND и DGND, Input для входов и т. д., — установите флажки в разделе Electrical Type. Перемещаться от вывода к выводу можно с помощью клавиш **Page Up** и **Down** на клавиатуре.
- Нажмите **ОК**, когда закончите.

Для назначения выводам номеров используем Property Assignment Tool. Для инициализации этого механизма выберите команду из раздела *Tools* основного меню и введите NUM=# в поле String и значение 1 в поле Count. Установите флажок On Click и закройте диалоговое окно с помощью кнопки **ОК**. Теперь внимательно щёлкните по каждому выводу в порядке возрастания их номеров (OUT, AGND, и т.д.). При каждом щелчке вывод будет пронумерован PAT. По завершении не забудьте остановить PAT, щёлкнув по кнопке **CANCEL** в его диалоговом окне.

Все, что нам осталось сделать — это создать устройство. Чтобы это сделать, выделите все выводы и основание (прямоугольник) — самый простой способ выделить все, использовать режим выделения (иконка на левой инструментальной панели с изображением курсора) и прочеркнуть мышкой все по диагонали. Теперь выберите команду Make Device из раздела Library основного меню, чтобы открыть диалоговое окно. Введите имя 7110 в поле Device Name и символ «U» в поле Reference Prefix. Затем нажимайте кнопку **Next>** пока не появится список перезаписываемых библиотек устройств, выберите подходящую библиотеку и щёлкните по кнопке **ОК**, чтобы сохранить новое устройство.

ПОСЛЕДНИЕ ШТРИХИ

Теперь, когда устройство 7110 определено, вы можете размещать его на чертеже, соединять с другими компонентами и вводить обозначения — используйте Automatic Annotator в режиме Incremental, чтобы обозначить новые элементы, не цепляя уже существующие обозначения.

Этикетки и скобка около шести входных выводов DA0-DA5 сделаны графическими средствами. ISIS предоставляет возможности для отрисовки линий, квадратов, окружностей, дуг и текста на вашем чертеже; все можно найти, как иконки, на левой инструментальной панели режимов.

Скобка состоит из трёх линий — выберите иконку *Line*, щёлкните в начале и щёлкните в конце линии. Поместите текст FROM PIO-1A, как показано, выбрав иконку *Text*; выделите полученный текст, поверните его на 90° против часовой стрелки, используя выпадающее меню, щелчок правой клавишей мышки, и перенесите в нужное место.

Окончательно вам нужно нарисовать рамку чертежа и оформить штамп. Для рамки выберите иконку *Box* на инструментальной панели, выберите просмотр таким, чтобы был виден весь чертёж и края листа (темно синие); в списке видов графических объектов выберите BORDER, щёлкните левой клавишей мышки по верхнему левому углу рамки и проведите прямоугольник для рамки (по умолчанию он может появиться залитым), ещё раз щёлкните левой клавишей мышки; подведите курсор мышки к краю прямоугольника и щелчком правой клавиши мышки вызовите выпадающее меню, где есть раздел *Edit Properties*, открывающий диалоговое окно свойств графического объекта; снимите флажок *Follow Global?* в пункте Fill Style и, используя

кнопку со стрелкой слева от окна, откройте список стилей и выберите *None*. Нажмите на кнопку **This Graphic Only**, чтобы получить только для рамки прозрачную заливку.

Штамп следует обсудить подробнее. Фактически он не отличается от любого другого символа, который вы можете использовать для фирменного логотипа. Есть раздел «*БЛОК ШТАМПА*», посмотрите его. И штамп можно нарисовать вручную с помощью общих графических средств. Но можно использовать штамп, предлагаемый ISIS: щёлкните правой клавишей мышки в рабочей области, выберите *Place->Symbol->From Libraries* и найдите в списке HEADER. В окне предварительного просмотра вы можете увидеть штамп с тем содержимым, которое вы использовали в диалоге свойств чертежа (*Design->Edit Design Properties*) и листа. Программа поддерживает в графическом тексте кириллицу.

СОХРАНЕНИЕ, ПЕЧАТЬ И ВЫВОД НА ПЛОТТЕР

Вы можете сохранить вашу работу в любое время, используя команду *Save* раздела *File* основного меню, и сейчас самое время это сделать! Опция *Save As* позволяет вам сохранить работу под другим именем файла, чем то, что было при загрузке чертежа.

Чтобы напечатать схему, вначале выберите правильный принтер, используя команду *Printer Setup* из меню *File*. Этим активизируется системный Windows диалог для выбора и конфигурации принтера. Детали зависят от конкретной версии Windows и драйвера принтера — обратитесь к документации на операционную систему и ваш принтер. Когда выбран правильный принтер, закройте диалог и выберите опцию *Print* меню *File*, чтобы начать печать. Печать можно прервать, если нажать клавишу **ESC** на клавиатуре, хотя может пройти некоторое время, прежде чем печать остановится — и ISIS, и ваш принтер должны опустошить их буферы.

Дальнейшие детали по работе с принтером и графическим выводом есть в разделе «*ПОЛУЧЕНИЕ ЖЕСТКОЙ КОПИИ*».

Если у вас демо-версия, заметьте, что вы можете печатать только не модифицированные примеры. Попробуйте это сейчас, загрузив файл из папки примеров выполнения разработки.

ПОДРОБНЕЕ О СОЗДАНИИ КОМПОНЕНТОВ

Создание много-элементных устройств

Сейчас мы определим подходящую часть библиотеки для счетверённого операционного усилителя TL074. Поскольку есть четыре отдельных ОУ в едином корпусе, наше руководство продемонстрирует вам, как создавать много-элементные устройства, используя *Visual Packaging Tool* (инструмент визуальной упаковки).

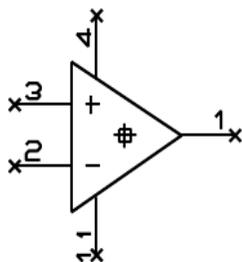


Рисунок слева показывает изображение нового ОУ до того, как оно сделано. Изображение создано с помощью графики, пяти выводов и origin marker (маркер оригинала). Рассмотрим два пути создания изображения операционного усилителя. Самый лёгкий путь — использовать предопределённый символ для ОУ. Выполните следующее:

- Щёлкните в свободном месте рабочей области правой клавишей мышки, чтобы получить выпадающее меню, где выберите *Place->Symbol->From Libraries*. Этим вы откроете диалоговое окно системной библиотеки, где в списке можно найти OPAMP.
- После выделения элемента, когда окно диалога закрыто, этот элемент появится в списке графических объектов в окне выбора (под *окном обзора*).
- Переместите указатель мышки на свободное место в *окне редактирования* и используйте левую клавишу мышки для добавления ОУ.

Теперь добавьте выводы, этот такой же процесс, как был использован ранее при создании аттенюатора 7110:

- Выберите иконку *Device Pins Mode*, чтобы получить список доступных типов выводов и выбрать Default тип.
- Используйте иконки *Rotate* и *Mirror* для правильной ориентации выводов, прежде чем размещать их на рисунке.
- Когда все выводы окажутся на месте, отредактируйте каждый из них, последовательно выделяя и с помощью диалога *Edit Pin* обозначьте выводы, задайте их электрический тип. Мы должны присвоить имена выводам, чтобы ссылаться на них в *Packaging Tool*, однако нам не нужно, чтобы они были видны на чертеже (использование выводов скрыто от графики), поэтому убедитесь, что флажок *Draw Name?* сброшен. Заметьте, что нужды в нумерации выводов нет, поскольку это будет выполнено при использовании *Packaging Tool*.

Выводы питания, названные V+ и V-, имеют электрический тип *Power*; если вы поместите их с левой стороны ОУ, вы обнаружите, что они только касаются скошенных сторон рисунка ОРАМР, сохраняя концы выводов (отмеченных, как «X») на квадратике сетки. В подобной ситуации, они не касаются линии, вы можете изменить шаг сетки и передвинуть их. Входные выводы, названные +IP и -IP, имеют электрический тип *Input*. Выходной вывод ОР типа *Output*.

Завершением будет размещение *Origin marker*. Выберите *Place->Marker->ORIGIN* из выпадающего меню (щелчок правой клавишей мышки), чтобы отобразить список системных символов. Щёлкните по ORIGIN, чтобы он появился в окне *выбранных объектов*, и поместите символ маркера в центр изображения операционного усилителя. *Origin marker* отображается как квадратик с перекрестием, а предназначен для индикации ISIS того, как должен появиться символ около курсора мышки, когда устройство перетаскивается или помещается на чертеже.

Сейчас мы закончили создание устройства. Выделите все части компонента — символ ОУ, выводы и *Origin marker* — с помощью мышки и вызовите команду *Make Device* из раздела *Library* основного меню. Затем продолжите следующим образом:

- Введите *Device Name* как TL074 и *Prefix* как «U».
- Щёлкните по кнопке **Next**, чтобы перейти к странице *Packaging*, где щёлкните по *Add/Edit*, чтобы запустить собственно *packaging tool* (инструмент упаковки).

Visual Packaging Tool

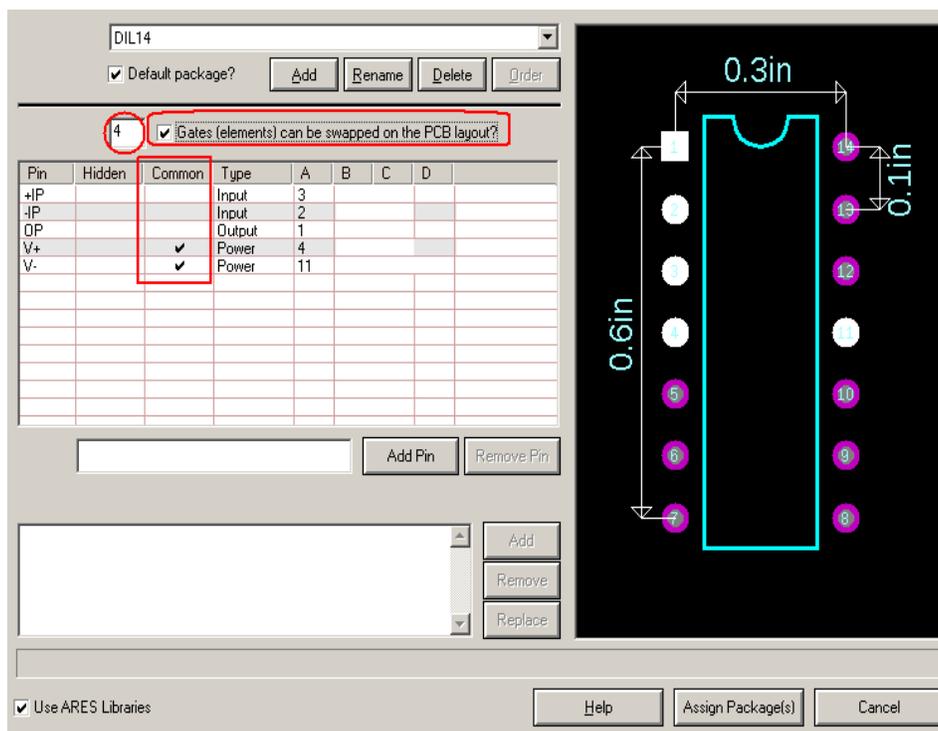
Этот инструмент предоставляет графическое окружение, с помощью которого элементу схемы назначается одно или более посадочное место (контактные площадки) при разводке печатной платы. Для каждого корпуса создаётся таблица соответствия номеров выводов и их имён так, чтобы разные корпуса могли иметь разные номера выводов для тех же выводов схемы.

Когда инструмент упаковки запущен, первое, что нужно сделать — это создать корпус:

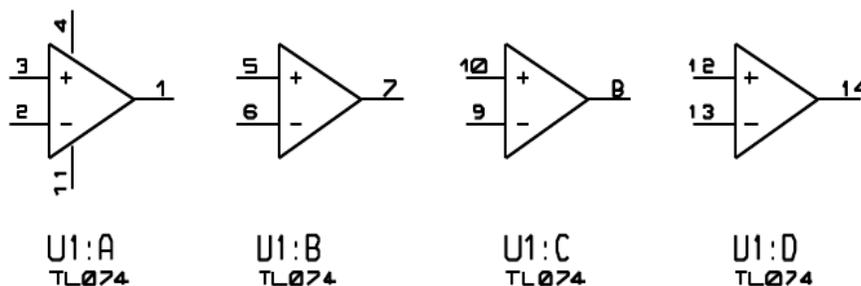
- Щёлкните по кнопке **Add**. Это запустит обозреватель библиотеки ARES.
- Введите в окне *Keywords*: DIL14.

Теперь вам нужно сделать следующие изменения в установках по умолчанию для корпуса:

- Измените количество элементов с 1 до 4. Это соотносится с тем фактом, что есть четыре операционных усилителя в одной физической DIL14 упаковке.
- Пометьте выводы V+ и V- как общие. Это означает, что они будут иметь те же номера выводов для всех ОУ корпуса, и что вы можете соединять питание с любым из этих выводов или со всеми выводами питания много-элементного компонента на схеме. Все эти соединения будут считаться взаимосвязанными.
- Установите флажок *Gates can be Swapped*. Это покажет, что элементы идентичны и ARES может выполнить операцию замены вентиляей.



Теперь назначим номера выводам. Фактическое расположение выводов, которое мы создаём, показано ниже:



Продолжите следующим образом:

- Щёлкните по колонке «A» для вывода +IP.
- Либо щёлкните по выводу «3» на рисунке корпуса или введите «3» с клавиатуры и нажмите клавишу **ТАВ** на клавиатуре. В любом случае вывод 3 будет подсвечен на рисунке корпуса, чтобы показать вас, что назначение принято и курсор переместится в строку -IP.
- Теперь повторите процесс для других номеров выводов, пока выводы корпуса не подсветятся все. И, конечно, благодаря визуальной подсказке, вы не пропустите ни одного вывода.

LABCENTER ELECTRONICS

И, наконец, щёлкните по кнопке *Assign Packages*, чтобы вернуться в помощника *Make Device*, и сохранить устройство в вашей USERDVC библиотеке, как это было сделано для 7110.

Есть, как вы, вероятно, поняли, большие функциональные возможности, встроенные в *packaging tool*. Обсуждение их и дальнейшие примеры вы найдёте в разделе «**БИБЛИОТЕКИ УСТРОЙСТВ**».

Создание похожих устройств

После того, как устройство TL074 определено, вы можете почти сразу определить TL064 и TL084. Просто поместите TL074, выделите его, и используйте команду *Make Device*; измените имя на TL064 (или другое) и сохраните. Что может быть проще? Если вам надо что-то добавить в базовую модель TL074, возможно, немного графики, вы можете просто добавить это поверх TL074 перед вызовом *Make Device*. И наоборот, если TL074 вполне вас устраивает, но вам нужно что-то подредактировать до того, как модель станет новым устройством, вы должны разложить TL074 на составные части, выделив все и вызвав команду *Decompose* из раздела *Library* основного меню; отредактировать и/или добавить что-то и затем создать новое устройство.

Замена компонентов в чертеже

Теперь вы можете заменить четыре операционных усилителя фильтра подходящими частями TL074. Чтобы это сделать, выберите TL074, как и остальные компоненты, убедитесь, что курсор мышки с контуром элемента покрывает тот, что вы хотите заменить, а один или несколько выводов покрывают старые и щёлкните левой клавишей мышки. Подтвердите, что нужно заменить компонент. ISIS перенесёт соединения старого элемента на новый, сохранив все обозначения старого компонента нетронутыми.

СИМВОЛЫ И БИБЛИОТЕКА СИМВОЛОВ

Выделите три линии, формирующие скобку около входов 7110. Затем вызовите *Make Symbol* из раздела *Library* основного меню, введите TEST в качестве имени символа и нажмите клавишу **ENTER**. Теперь выберите иконку *2D Graphics Symbols Mode*. Вы увидите, что элемент TEST появился в *окне выбора*. Щёлкните по нему и попробуйте добавить в чертёж. Наиболее общее применение этого находят такие элементы, как ОРАМР, которые нужны для многих типов устройств, логотипов, эмблем и т. п.

Есть одно обособленное использование символов — вставка в штамп (HEADER block). Предопределённый символ был создан из линий, прямоугольника и некоторых специальных текстовых элементов, которые автоматически заменяются свойствами, ассоциированными с текущим чертежом и листом. Например, текстовый объект со строкой:

@DTITLE

автоматически появится в качестве заголовка чертежа, введённого в диалоговое окно *Edit Design Properties*. Полный список ключевых слов представлен в разделе «**ШТАМП**».

СОЗДАНИЕ ОТЧЕТОВ

Теперь, когда рисунок завершён, вы можете сгенерировать Netlist, Bill of Materials (BOM) и Electrical Rules Check (ERC). Каждый отчёт создаётся вызовом соответствующей команды из раздела Tools основного меню. Выходной отчёт отображается во всплывающем окне обозревателя текста, из которого его можно сохранить в файл, выбрав кнопку **Save as** или скопировать в буфер обмена для использования в других пакетах с помощью кнопки **Copy to clipboard**; кнопка **Close** очистит окно отчётов и вернёт вас в редактор. Заметьте, что последний отчёт или запись, созданная в журнале симуляции, обслуживаются ISIS: чтобы вновь увидеть их выберите команду *Text Viewer* в разделе *System* основного меню.

Отчёт Bill of Materials должен быть понятен, хотя вы можете познакомиться с ним подробнее в разделе «Bill Of Materials».

Отчёт Electrical Rules Check будет содержать несколько ошибок, поскольку учебный пример — не вполне завершённая разработка: особенно то, что вывод VBB 7110 помечен как свободный, о чем вполне можно было забыть в реальной ситуации.

Дальнейшая информация об отчётах дана в разделе «Создание отчётов», включая рассказ об особенностях получения и использования Netlist. Для тех, у кого есть и ISIS, и ARES, и кто намерен их использовать вместе, есть раздел «Isis и Ares».

БОЛЬШИЕ ПРОЕКТЫ

В этом последнем разделе мы хотели бы обратить внимание на полуготовый проект EPE.DSN. Это многолистная иерархическая схема для управляемого микропроцессором EPROM программатора/эмулятора (EPE). Таким образом, проект представляет существенную часть электроники на более глубоком уровне сложности, чем тот, что вы, возможно, ожидали от использования системы ISIS.

Проект EPE разбит на три листа формата A3 (Processor, Emulator и PSU). Подсхемы использованы для представления банка Emulation RAM (которых здесь 4, что даёт совместимость с 32 битовой симуляцией) и Programmable Power Supply (PPSU), которых нужно 6 для работы с выходами EPROM 27 серии.

Загрузите проект в ISIS: используйте команду из набора *File* основного меню, а затем выберите EPE.DSN в проводнике. Вы можете найти проект в папке «Samples\Schematic & PCB Design» вашей установки Proteus. В качестве альтернативы вы можете увидеть любые примеры разработки с помощью выбора SAMPLES в подсказке раздела *Help*.

Первый лист — это CPU. Просмотрите его, используя обычные возможности панорамирования и увеличения. Затем, чтобы увидеть остальную часть проекта, используйте команду *Goto Sheet* из *Design* меню. Выберите второй пункт в дереве проекта и в рабочее поле будет загружен лист Emulator Control. Можете уменьшить вид, чтобы увидеть все. 4 большие квадрата — это подсхемы. Этикетки в верхней части это ID подсхем (как обозначение частей микросхем).

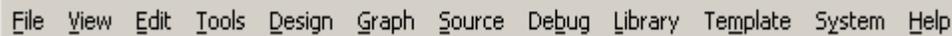
Вы можете войти в подсхему иначе: выделите её и, щёлкнув правой клавишей мышки, выберите из выпадающего меню *Goto Child Sheet* или нажмите клавиши **CTRL+C**. ISIS заменит лист Emulator Control на ERAM bank. Рассмотрите схему ERAM bank, в особенности обратите внимание на номера нескольких компонентов. Чтобы вернуться назад, нажмите **CTRL+X**.

Войдите в другой ERAM bank и сравните номера компонентов этой схемы с предыдущей — хотя оба образца подсхем имеют одинаковые электрические цепи (если вы внесёте изменения в один из них, это тотчас отобразится на остальных, что упрощает работу над проектом), и каждая имеет собственные обозначения компонентов; это работа *Design Global Annotation*.

Теперь, когда вы знаете, как загружать разные листы, как «бродить» по иерархии проекта, вы можете просмотреть остальную часть проекта EPE. Это прекрасная смесь аналоговых, цифровых и микропроцессорных цепей, которая показывает, как ISIS великолепно подходит для работы с разного типа схемами.

ВИД ЭКРАНА

Основное меню



Основное меню занимает верхнюю часть окна и служит для общих целей (выбор команд из меню), как и любое другое приложение Windows. Помимо того, выше основного меню есть область, показывающая имя загруженного файла.

Инструментальные панели

Как и другие современные приложения Windows, ISIS предлагает доступ к ряду команд и режимов через иконки инструментальных панелей. Инструментальные панели могут перетаскиваться к любому краю окна ISIS.

Инструментальная панель команд

Панель, расположенная ниже основного меню (по умолчанию) предлагает альтернативный доступ к командам основного меню, как следующие:

Работа с файлами/Печать



Команды отображения



Команды редактирования



Инструменты разработки



Если вы работаете за относительно маленьким монитором, вы можете скрыть любую или все инструментальные панели, используя *Toolbars* раздела *View* основного меню.

Панель выбора режима

Панель, расположенная по левому краю экрана предназначена к выбору режима редактора, то есть, определяет поведение *окна редактирования* при щелчке мышки в рабочем поле.

Основные режимы



Гаджеты



Графика



Заметьте, что панель основных режимов нельзя скрыть, поскольку её функции не дублируются в основном меню.

Панель ориентации

Панель ориентации отображает и управляет поворотами и отражениями объектов, размещённых в рабочей области.

Повороты



Отражения



Окно поворотов позволяет ввести угол, но в ISIS используются только ортогональные углы.

Окно редактирования

Окно редактирования отображает часть схемы, которую вы в настоящий момент редактируете. Содержание *окна редактирования* может перерисовываться, если использовать команду *Redraw* раздела *View* основного меню. Этим так же перерисовывается *Overview Window* (окно обзора). Вы можете использовать этот механизм после любых других команд, если вид чертежа становится несколько неопрятным.

Панорамирование

Вы можете перемещать *окно редактирования* по другим частям чертежа несколькими путями:

- Щёлкнув левой клавишей мышки по *окну обзора* — это сместит *окно редактирования* от центра.
- Перемещая курсор мышки по *окну редактирования*, когда вы удерживаете клавишу **SHIFT** нажатой и касаетесь курсором краев *окна программы*. Этим панорамируется изображение в соответствующем направлении. Мы обозначили этот приём, как *Shift-Pan*.

- Указав место в окне редактирования и нажав клавишу **F5**. Этим изображение смещается относительно позиции курсора.
- Используя иконку панорамирования инструментальной панели.

Увеличение/Уменьшение

Вы можете увеличить или уменьшить изображение, используя команды *Zoom In* и *Zoom Out*, которые можно вызывать и горячими клавишами **F6** и **F7** (и колёсиком мышки). Нажатием на клавишу **F8** вы вернёте изображение всего листа. Вы можете также использовать соответствующие иконки инструментальной панели.

Если для выполнения команд используется клавиатура, тогда *окно редактирования* перерисовывается к положению вокруг курсора мышки, которое он занимал перед этим. Это также дает возможность панорамировать нажатием на клавишу **F5** для текущего уровня с немедленным перемещением мышкой туда, где вы хотели бы иметь центр нового изображения.

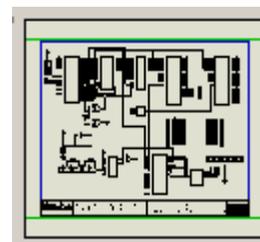
Переменное масштабирование

Произвольная степень масштабирования может быть получена при использовании механизма *Shift-Zoom*. Нужная область рисунка может быть выделена так, чтобы заполнить окно редактирования, путём нажатия и удержания клавиши **SHIFT**, нажатия на левую клавишу мышки и протаскивания её по нужной области чертежа. Область чертежа может отмечаться и в *окне редактирования*, и в *окне обзора*.

Вы можете масштабировать вид и щелчком по иконкам инструментальной панели.

Окно обозревателя

Это окно обычно показывает упрощённое представление всего чертежа и имеет полудюймовую сетку. Синим цветом обозначен контур листа, тогда как зелёным — область, в настоящее время видимая в *окне редактирования*.



Щелчок левой клавишей мышки позволяет панорамировать и перерисовывает вид окна редактирования.

В иное время окно обзора используется для отображения вида объекта, который выбран для размещения на чертеже. Этот *предварительный просмотр* активизируется при следующих обстоятельствах для любых объектов, которые могут быть ориентированы:

- Когда объект выбран в селекторе объектов.
- Когда выполняется поворот или отражение иконками.
- Когда выбрана иконка типа объекта для тех, чья ориентация может задаваться (то есть, иконка компонентов, иконка выводов устройства и т. д.).

LABCENTER ELECTRONICS

Окно предварительного просмотра автоматически очищается всякий раз, когда вы размещаете объекты или когда вы выполняете переориентацию объекта иную, чем перечисленные выше.

Ширина и высота *окна обзора* может меняться — достаточно перетащить его края. Если вы переместите правый край так, что он дойдёт до другого края приложения, *окно обзора* и *окно выбранных объектов* будут помещены справа.

741 Chain Sample

Here is a simple circuit of several cascaded op-amps that can be used to put the PROSPICE simulator through its paces. The 741's are modelled with a standard macro model.

labcenter
Electronics

741 Chain Sample

Labcenter Electronics, 53-55 Main Street, Grassington, North Yorkshire, BD23 5AA
Fax: +44 (0)1756 752857 Tel: +44 (0)1756 753440
Email: info@labcenter.co.uk WWW: http://www.labcenter.co.uk/

P	L	DEVICES
		555
		02013A1R2CAT2A
		4009
		4023.IEC
		74107
		74122
		PIC16F628A

Окно выбранных объектов

Окно выбранных объектов используется для выбранных компонентов, генераторов, типов графиков и т. д. На нем всегда появляется этикетка, показывающая, что есть в списке, а, кроме того, она используется как дополнение для отображения состояния панели иконок, показывая текущий режим работы.

Ширина и позиция *окна выбора* может подстраиваться вместе с *окном обзора*, как это описано выше.

Кнопки в верхнем левом углу *окна выбранных объектов* активируют разные функции, такие как вызов обозревателя библиотек и выбора компонентов. Выполняемые функции зависят от текущего режима работы редактора.

P	L	DEVICES
		555
		02013A1R2CAT2A
		4009
		4023.IEC
		74107
		74122
		PIC16F628A

Отображение координат

Текущие координаты указателя мышки отображаются в правом нижнем углу рабочего окна. Отображение может быть в условных и метрических единицах, и можно задать точку отсчёта. Более подробно это описано в разделе СИСТЕМА КООРДИНАТ.

Команда *X-Cursor* (*View* основного меню) отобразит маленькое или большое перекрестие дополнительно к стрелке курсора, определяя точное местоположение текущих координат. Это особенно полезно в сочетании с *Real Time Snap*, поскольку даёт вам непосредственное представление о том, что «думает» программа о вашем местонахождении.

СИСТЕМА КООРДИНАТ

Все координаты в ISIS фактически в единицах 10 нм, чисто для совместимости с ARES. Однако координаты читаются в 1 thou единицах (одна тысячная дюйма). Исходной точкой считается центр рабочей области, так что используются и положительные, и отрицательные значения координат. Координаты курсора отображаются в нижней части рабочего окна.

Увеличение происходит на 1 thou.

Точка отчёта

Хотя команда *Origin* есть в разделе *View*, её следует использовать только через горячую клавишу (**O**). Её задача обнулить координаты в текущей позиции курсора; возможность, которая полезна при работе со сложными образцами площадок, задаваемых набором размеров на рисунке компонента.

Когда точка отсчёта задана, координаты отображают это изменением цвета с чёрного на пурпурный, чтобы напомнить об этом.

Чтобы сбросить режим изменения точки отсчёта, дайте команду *Origin* ещё раз.

Сетка

Сетка отображается в окне редактирования и может быть включена и выключена командой *Grid* в разделе *View* основного меню. Расстояние шага отображает текущее задание сетки, пока это не становится неудобным при увеличении расстояния между точками.

Привязка к сетке

Вы можете заметить, что когда указатель в окне редактирования, увеличение координат отображается фиксированным шагом — начально в 100. Это называется привязкой и помогает вам располагать компоненты другие объекты чётко по сетке. Шаг сетки может быть выбран командой *Snap* раздела *View* основного меню или непосредственно клавишами **F4**, **F3**, **F2** и **CTRL+F1**.

Если вам нужно увидеть, где точно находится данное место, вы можете использовать команду

X-Cursor (View), что отобразит либо маленькое, либо большое перекрестие в этом месте.

Real Time Snap

Более того, когда указатель находится около вывода или провода, позиция курсора будет привязана к этим объектам. Такая функция называется *Real Time Snap* и позволяет вам соединять выводы и провода, которые не находятся в текущем шаге сетки. Вы можете переключать эту функцию, используя команду *Real Time Snap* из раздела *Tools* основного меню или клавишами **CTRL+S**.

С очень большими чертежами на медленных компьютерах *Real Time Snap* может стать причиной некоторого отставания курсора от указателя. В этих условиях, возможно, следует отказаться от этой функции.

КОМАНДЫ ХРАНЕНИЯ

ISIS создает следующие типы файлов:

Файлы проекта	(.DSN)
Файлы резервирования	(.DBK)
Файлы секций	(.SEC)
Файлы модулей	(.MOD)
Файлы библиотеки	(.LIB)
Файлы спецификации, Netlist	(.SDF)

Файлы проекта содержат всю информацию об одной схеме и имеют расширение файла «DSN». Предыдущие версии ISIS использовали расширения ISS, IDS и IWS; они могут конвертироваться автоматически, подразумевается, что файлы конвертеров IDSCVT40.DLL и/или IWSCVT40.DLL установлены. Резервные копии файлов проекта создаются, когда сохраняется существующий файл, и получают расширение «DBK».

Секции чертежа могут экспортироваться в файл секции и позже прочитываться в другой чертеж. Файлы секций имеют расширение «SEC». Их можно прочитать и записать командами *Import* и *Export* раздела *File*.

Файлы модулей имеют расширение «MOD» и используются в сочетании с другими возможностями иерархической разработки. Больше информации вы найдете в главе «Иерархический проект».

Библиотеки символов и устройств имеют расширение «LIB».

Файлы спецификации Netlist создаются, когда экспортируется связность в ProSPICE и ARES, с расширением «SDF»; другие расширения применяются при создании файлов netlist в формате программ других производителей.



Система симуляции Proteus VSM вполне успешно использует файлы других типов. Детали вы найдете в руководстве к Proteus VSM.

Начиная новый проект

Команда *New Design* очистит все существующие данные проекта и даст единственный пустой лист формата A4. Имя разработке даётся UNTITLED.DSN и это же имя будет использовано командой *Save Design*, а также в качестве предопределённого имени в других селекторах файлов.

Загрузка проекта

Проект может быть загружен несколькими способами:

- Из командной строки: ISIS <my_design>
- Командой *Open Design...*, когда программа ISIS работает.
- Двойным щелчком по файлу в Windows Explorer.

Сохранение проекта

Вы можете сохранить свой проект при выходе из ISIS с помощью команды *Exit*, или в любое время, используя команду *Save Design*. В обоих случаях проект сохранится под тем именем, под которым он был загружен. Предыдущая версия получит текстовый префикс «Backup of».

Команда *Save As* позволяет вам сохранять проект в другом файле.

Импорт/Экспорт секции

Команда *Export* из *File* создаёт выходной файл секции всех текущих выделенных объектов. Файл впоследствии может быть прочитан и добавлен на другой лист с помощью команды *Import*. После выбора файла секции операция идентична функции *Block Copy*. Эти команды ничего не затрагивают экспорт графиков в DTP пакет. Команда *Export Graphics* восполняет пробел.

Выход из ISIS

Когда вы решили закончить сессию ISIS, вы можете использовать команду *Exit* из раздела *File* основного меню или клавишу **Q**. Если проект был изменён, вы получите приглашение сохранить проект.

ОБЩИЕ ВОЗМОЖНОСТИ РЕДАКТИРОВАНИЯ

Размещение объектов

ISIS поддерживает множество типов объектов, а полный набор свойств и поведения каждого из типов дан в следующих главах. Однако основные шаги по размещению объектов одинаковы для всех типов.

Чтобы поместить объект:

1. Выберите подходящую иконку на инструментальной панели *Mode* для категории объекта, который вы хотите поместить на схему.
2. Если тип объекта *Component*, *Terminal*, *Pin*, *Graph*, *Symbol* или *Marker* выберите имя объекта, который вам нужен в *окне выбора объекта*. Для объектов *Component*, *Terminal*, *Pin* и *Symbol* может потребоваться вызов менеджера выбора из библиотек.
3. Если объект может быть ориентирован, он появится в *окне обзора*. Вы должны приспособить его ориентацию к нужному виду, используя иконки *Rotation* и *Mirror*.
4. В завершение щёлкните по окну редактирования, чтобы поместить или перетащить объект. Точные процедуры могут варьироваться для каждого из типов объектов, но они интуитивно понятны и похожи на работу с другими графическими приложениями. Более детально все описано в главе «Спецификация объектов».

Выделение объекта

Любой объект может быть выделен правым щелчком мышки. Этим объект подсвечивается и выбирается для последующих операций редактирования.

- Все провода, приходящие к выделенному объекту, будут выделены.
- Группа объектов может быть выделена либо последовательным щелчком с удержанием клавиши *Ctrl*, либо отрисовкой области выделения (нажать и удерживать левую клавишу мышки и очертить прямоугольную область выделения). Выделены будут только объекты, полностью попавшие в область выделения.
- Выделение со всех объектов можно снять, щёлкнув мышкой на свободном месте рабочего поля.

Удаление объекта

Вы можете удалить любой объект, щёлкнув по нему правой клавишей мышки и выбрав из выпадающего меню *Delete Object*. Все провода, подходящие к объекту, будут тоже удалены, исключая случай, когда в точке соединяются два провода. В этом случае провода соединятся.

Перетаскивание объекта

Вы можете перетащить (то есть, перепозиционировать) любой выделенный объект: щёлкните по нему, подцепите его и перетащите в новое место мышкой, нажав и удержав левую клавишу. Это относится не только к целым объектам, таким как компонент, но также касается и их этикеток.

- Если функция *Wire Auto Router* включена, и есть провода, подходящие к объекту, тогда они будут переразведены или упорядочены. Это может занять некоторое время (10 секунд или около того), если объект имеет много подключённых проводов; возле указателя появляются и остаются песочные часы, пока процесс не закончится.
- Если вы перетащили объект по ошибке, а все соединения стали ужасающе неправильны, используйте команду *Undo* или нажмите клавишу *U*, чтобы вернуть все в первоначальный вид.

Перетаскивание этикеток объекта

Многие типы объектов имеют одну или больше этикеток, прикреплённых к ним. Например, каждый компонент имеет обозначение и номинал. Чтобы улучшить вид чертежа, можно легко переместить эти этикетки.

Чтобы переместить этикетку:

1. Выделите объект, щёлкнув по нему (или по этикетке).
2. Указатель мышки поместите на надпись и нажмите левую клавишу мышки.
3. Перетащите надпись в нужное место. Если вам нужно позиционировать этикетку очень

точно, вы можете изменить шаг сетки (используя клавиши **F4, F3, F2, CTRL+F1**), даже во время перетаскивания.

4. Отпустите клавишу мышки для завершения перетаскивания.

Изменение размеров объекта

Можно изменить размеры подсхем, графиков, линий, прямоугольников и окружностей. Когда вы выделяете эти объекты, маленькие чёрные квадратики окаймляют их; и размер объекта можно изменить, перетаскивая края за эти «ручки».

Чтобы изменить размеры объекта:

1. Выделите объект щелчком левой клавиши мышки.
2. Если объект может менять размер, по контуру выделения появятся маленькие черные квадратики.
3. Измените размер объекта, поместив курсор на такой квадратик (курсор изменит вид), нажмите и удержите левую клавишу мышки, и перетащите край в нужное место. «Ручки» исчезают во время перетаскивания так, чтобы не затенять сам объект.



См. раздел «Изменение размеров графики», где больше информации об использовании возможности изменения размеров графических объектов.

Изменение ориентации объекта

Многие типы объектов можно переориентировать — это и повороты на 0°, 90°, 270° и 360°, и отражения по осям x и/или y.

Чтобы изменить ориентацию объекта:

1. Щёлкните правой клавишей мышки по объекту.
2. Из выпадающего меню выберите нужную операцию *Rotation* или *Mirror*.

Иконки *Rotation* и *Mirror* на инструментальной панели работают для выбора ориентации вновь размещаемых объектов до того, как вы перенесёте их на чертёж.

Редактирование объекта

Многие объекты имеют графические и/или текстовые свойства, которые можно редактировать с помощью диалоговых форм, и, поскольку это достаточно общая операция, мы опишем несколько способов, как сделать это.

Для редактирования единственного объекта с помощью мышки:

1. Выделите объект, щёлкнув по нему левой клавишей мышки.
2. Щёлкните по нему левой клавишей мышки ещё раз.

Редактирование объекта с доступом к специальным режимам:

1. Выделите объект.
2. Нажмите **CTRL+E**.

Для текстовых элементов это вызовет появление встроенного редактора текста. А, если под курсором мышки нет объекта, не приведёт ни к каким действиям.

Для редактирования компонента по имени:

1. Нажмите клавишу **E**.
2. Введите обозначение (ID) компонента.

Этим вы локализуете объект и вызовете диалоговое окно свойств для любого компонента чертежа, и не только на текущей странице. После редактирования экран будет перерисован, а компонент окажется в центре. Вы можете использовать эту команду для локализации компонента даже в том случае, когда не намерены редактировать его.



Детали, относящиеся к диалоговым окнам, ассоциированным с каждым из объектов, даны в разделе «Спецификация объектов».

Редактирование этикеток объектов

Этикетки component, terminal, wire и bus могут все редактироваться во многом так же, как и объекты.

Для редактирования этикетки единичного объекта с помощью мышки:

1. Выделите объект, щёлкнув по нему левой клавишей мышки.
2. Щёлкните по этикетке, как для перетаскивания, но сразу отпустите клавишу без перемещения мышки.

Копирование всех выделенных объектов

Для копирования секции схемы:

1. Выделите требуемые объекты либо индивидуально, либо группой с помощью мышки.
2. Щёлкните по иконке *Block Copy*.
3. Перетащите копию в нужное место и щёлкните левой клавишей мышки.

4. Повторите шаг 3, если нужно несколько копий.
5. Щёлкните правой клавишей мышки, чтобы завершить копирование.

Когда компоненты скопированы, их обозначения автоматически сбрасываются к состоянию «без обозначения», чтобы подготовить их к автоматической аннотации и предотвратить появление множества образцов с одинаковыми ID (идентификаторами).

Перемещение выделенных объектов

Чтобы переместить несколько объектов:

1. Выделите нужные объекты либо индивидуально, либо группу с помощью прямоугольного выделения.
2. Щёлкните по иконке *Block Move*.
3. Перетащите блок в нужное место и щёлкните левой клавишей ещё раз.

Поведение проводов в процессе перемещения блока несколько хитро. ISIS переместит все провода и части проводов, заключённые в выделение, без их перерисовки, а затем, там, где провода пересекают границы выделения, они будут перерисованы от последней точки внутри выделения к первой точке вне границы выделения. Из этого следует, что вы можете контролировать, будет ли секция соединений сохранена или будет перерисована, согласно тому, где она окажется — внутри или вне выделенной области.



Вы можете также использовать механизм перемещения блока для перетаскивания секции соединений, но без перемещения каких-либо объектов. Более подробно это описано в главе «Перетаскивание соединений».

Удаление всех выделенных объектов

Чтобы удалить группу объектов:

1. Выделите требуемые объекты либо индивидуально, либо всю группу с помощью прямоугольного выделения, как описано в разделе «Выделение объектов».
2. Щёлкните по иконке *Delete*.

Если при этом вы что-то удалили по ошибке, вы можете все исправить, используя команду *Undo*.

ПРОВЕДЕНИЕ СОЕДИНЕНИЙ

Размещение проводов

Вы могли обратить внимание на отсутствие иконки *Wire*. Причина этого в том, что ISIS достаточно разумная программа, чтобы обнаружить, когда вам требуется соединение. Это избавляет от скучного выбора режима проводки.

Чтобы провести соединение между двумя объектами:

1. Щёлкните левой клавишей мышки по точке соединения первого объекта.
2. Если вы хотите, чтобы ISIS автоматически провёл проводку, просто щёлкните по второй соединяемой точке. С другой стороны, если вы хотите самостоятельно определить, как пойдёт провод, вы можете щёлкнуть левой клавишей мышки на промежуточной точке соединения, которая станет угловой в соединительном проводе.

Точка соединения может принадлежать только одному проводу. Для компонентов и разъёмов есть точки соединения на каждом выводе. Точка соединения (junction dot) имеет четыре точки соединения около её центра, так что к точке соединения можно присоединить четыре провода.

Поскольку очень часто нужно создать соединение с существующим проводом, ISIS также обрабатывает провода, как последовательность точек соединений. Более того, как таковое, если встречается соединение трёх проводов в одной точке, это непременно означает, что будет добавлена точка соединения. Этим полностью устраняется неопределённость, которая в противном случае может приводить к пропуску точек соединения.

Вы можете прервать проводку соединений, нажав клавишу **ESC**, на любой стадии процесса.

Wire Auto-Router

Wire Auto-Router (WAR) избавляет вас от рутины вырисовывания каждого провода. Функция включена по умолчанию, но может быть отменена двумя способами.

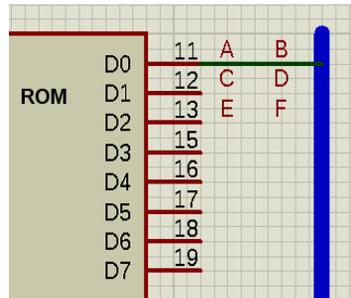
Если вы просто щёлкните левой клавишей мышки по двум точкам соединения, WAR попытается выбрать удобный способ для прокладки провода. Если же вы щёлкните по одной точке соединения, а затем щёлкните в любом месте без точки соединения, ISIS решит, что вы хотите вручную провести разводку и даст вам возможность щёлкать на всех поворотах на пути провода. И соединение завершится, когда вы окончательно щёлкните по второй точке соединения.

WAR можно полностью отключить, используя команду *Wire Auto-Router* в разделе *Tools* основного меню. Это полезно, если вы хотите провести диагональный провод непосредственно между двумя точками соединения.

Повтор соединения

Представьте, что вы соединяете шину данных 8-битового ROM с общей шиной данных на

чертеже и что вы уже разместили ROM, шину и входы шины, как показано ниже.



Вначале вам нужно щёлкнуть слева от A, затем B, чтобы провести горизонтальное соединение между ними. Но щёлкнув дважды по C вы вызовете функцию *Wire Repeat*, которая соединит точки C и D. Двойной щелчок по E соединит E и F, и т.д.

Wire Repeat в точности копирует предыдущий способ соединения. Если предыдущее соединение было сделано автоматически, тогда и последующие соединения будут сделаны автоматически. С другой стороны, если предыдущее соединение было проведено вручную, тогда его точное прохождение будет считано и использовано для нового проводника.

Перетаскивание проводников

Хотя проводники следуют общей схеме выделения и перетаскивания, есть несколько специальных приёмов, которые вы можете использовать. В частности:

- Если вы «подцепите» угол и «поташите» его, тогда угол будет просто следовать за курсором.
- Если вы «подцепите» середину горизонтального или вертикального сегмента соединения, сегмент будет перемещаться вертикально или горизонтально соответственно, а смежные сегменты «подтянутся», чтобы сохранить соединение.
- Если вы «подцепите» середину диагонального сегмента, или конец провода, тогда будет создан угол и затем соединение будет перемещено. Отметьте для последующей работы, что объект, с которым провод соединяется, не должен быть выделен, иначе ISIS будет думать, что вы хотите перетащить объект.
- Также возможно перемещать сегменты проводника и группы сегментов, используя команду *Block Move*.

Чтобы переместить сегмент или группу сегментов соединения:

1. «Очертите» сегмент(ы), которые вы хотите переместить. Вполне допустимо, чтобы прямоугольник выделения был линией, лежащей на единственном сегменте, если это удобно.
2. Щёлкните по иконке *Move*.

3. Переместите выделение в направлении ортогональном сегменту соединения, как показано на рисунке.
4. Щёлкните левой клавишей мышки, чтобы завершить процесс.

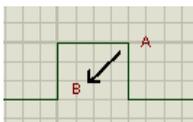
Если что-то получилось не так, используйте команду *Undo*, чтобы вернуться в исходное положение.



Следующие приёмы дают возможность быстро ликвидировать нежелательные петли в соединениях; возможно, они появились вокруг объектов, которые впоследствии были перемещены.

Чтобы удалить петлю из проводника:

1. Выделите провод, с которым собрались работать.
2. Щёлкните по одному углу петли и нажмите левую клавишу мышки.
3. Перетащите угол так, чтобы петля перекрыла себя, как на рисунке ниже.
4. Отпустите левую клавишу мышки. ISIS удалит петлю из соединения.



АВТОМАТИЧЕСКИЙ АННОТАТОР

ISIS может автоматически выбирать обозначения компонента для всех или некоторых компонентов на чертеже — этот процесс называется Auto-Annotation.

Эта функция инициализируется командой *Global Annotator* в разделе *Tools* основного меню.

Заметьте, что *Global Annotator* не может работать с гетерогенными много-элементными компонентами. Это происходит с несколькими катушками реле и контактами, например, поскольку не понятно, что с чем идёт.

Расстановка значений

Эта возможность предназначена для использования там, где программа анализа вычисляет некоторые значения для стандартных электрических цепей, а вы хотите импортировать их. Пример блока:

```
VALUES  
R1,10k  
C1,100n  
END
```

Значение резистора R1 будет 10 кОм, а C1 100 нФ.

РАЗНОЕ

Рамка чертежа

Когда создаётся новый лист либо в качестве первого листа схемы, либо при использовании команды *New Sheet*, он создаётся с размером, установленным в диалоге *Set Sheet Sizes* раздела *System* основного меню. Границы листа показаны темно-синим контуром, но это не отражается на твёрдой копии.

Если требуется рамка чертежа для окончательного вывода, вы должны поместить рамку поверх границ листа.

Команда *Set Sheet Sizes* — предмет дальнейшего обсуждения, поскольку служит двум целям:

- Можно изменить размер текущего листа, просто выполнив команду и установив флажок для задания нужного размера.
- Можно переопределить размеры листа, подсветить нужные данные заполняемых полей и ввести новые размеры. Если изменить размеры текущего листа, то изменения немедленно вступят в силу, но не коснутся других листов проекта. Чтобы изменить их все, вы должны последовательно вызвать *Set Sheet Sizes* и щёлкнуть **ОК** для каждого из них.

Диалог *Set Sheet Sizes* позволяет вам задать размеры в пяти стандартных позициях (от A4 до A0), равно как и в не стандартных, и выбрать размер текущего листа.

Для каждого размера листа (A4-A0 и пользовательский) есть два редактируемых поля. Левое поле определяет ширину (размер по x), а правое высоту (размер по y). Когда ISIS загружает лист, не принадлежащий к стандартным (A4-A0), он помещает размеры листа в поле User и задаёт размеры, как предопределённые, для загружаемого листа.

Предустановленные значения для стандартных размеров справедливы для большинства принтеров, но, пожалуйста, заметьте, что все принтеры имеют кромки, на которых они не могут работать — если вы собрались печатать что-то на этих кромках, вы можете не увидеть этого. Размеры технических кромок, где принтер не может печатать, варьируются от принтера к принтеру, и наилучший способ определить их — напечатать какой-нибудь чертёж с рамкой,

определяющей границы выбранного вами размера. Если один или более краёв не будет напечатан, значит выбранный вами размер листа превышает возможности принтера, и нужно уменьшить его.

Есть ещё одно, что нужно отметить касательно размеров листа: каждый лист в проекте несёт свои актуальные размеры с собой в файле проекта. Если проект загружается в копию ISIS с отличной конфигурацией размеров листа, это не будет иметь никакого влияния пока не будет команды *Set Sheet Sizes*.

Штамп

Обычная практика — на каждом листе чертить штамп, который показывает такие детали, как название проекта и листа, версию и количество страниц, и автора проекта.

В том случае, если вы хотите полностью контролировать представляемую информацию, штамп определён как элемент библиотеки символов, названный *HEADER*. Это выполняется обычным образом: размещением графического объекта, выделением его и вызовом команды *Make Symbol*. Важно, чтобы, если требуется название текущего проекта, был помещён текстовый объект со строкой *@DTITLE*, что автоматически приведёт к замене названия проекта при отображении и печати. Полный список таких ключевых слов приведён ниже.

- @DTITLE** Название проекта, полученное из диалога *Edit Design Properties*.
- @STITLE** Название листа, полученное из диалога *Edit Sheet Properties*. (*He Sheet Name*).
- @DOCNO** Номер документа проекта, полученный из диалога *Edit Design Properties*.
- @REV** Номер версии проекта, полученный из диалога *Edit Design Properties*.
- @AUTHOR** Имя автора проекта, полученное из диалога *Edit Design Properties*.
- @CDATE** Дата создания проекта, получаемая автоматически и в фиксированном формате.
- @MDATE** Дата изменения проекта, получаемая автоматически в фиксированном формате.
- @WS_CDATE** Дата создания проекта, автоматически получаемая в Windows «short date format».
- @WL_CDATE** Дата создания проекта, автоматически получаемая в Windows «long date format».
- @WS_MDATE** Дата изменения проекта в Windows «short date format».
- @WL_MDATE** Дата изменения проекта в Windows «long date format».
- @CTIME** Время создания проекта, автоматически получаемое в Windows «time format».
- @MTIME** Время изменения проекта, автоматически получаемое в Windows «time format».
- @PAGENUM** Текущий номер страницы в проекте.
- @PAGECOUNT** Общее количество страниц в проекте.
- @PAGE** Номер страницы в проекте с расширением X/Y (номер листа/общее количество).
- @FILENAME** Текущее имя файла проекта.
- @PATHNAME** Полный путь к файлу проекта.

LABCENTER ELECTRONICS

Формат длинной и короткой форм даты и времени Windows устанавливается в панели управления Windows в разделе локализации (язык и дата).

Заметьте, что ключевые слова должны появляться в начале текстовой строки, и строка не должна содержать дополнительный текст. Например, не помещайте текст в форме:

AUTHORED BY @AUTHOR ON @WS_MDATE

поскольку это не будет работать! Чтобы выполнить вышенаписанное, вам нужно написать четыре строки одна возле другой (AUTHORED BY, @AUTHOR, ON и @WS_MDATE).

С помощью графических объектов и такого специального текста можно определять штампы любого рода, какой вам нужен. В частности, вы можете вставить в штамп логотип вашей компании. Однажды определённый, штамп может размещаться на каждом листе проекта подобно любому другому графическому символу.

На задний план/На передний план

Временами некоторые объекты (особенно графика) перекрываются и становится трудно выделить тот, что вам нужен. По умолчанию ISIS берет объекты недавно размещённые, но вы можете реорганизовать последовательность, используя команды *Send to Back* и *Bring to Front*. Каждая команда оперирует с текущими выделенными объектами.

Вы также можете использовать эти команды для определения порядка на рисунке, когда создаёте новые устройства. Например, если вы создаёте новый символ операционного усилителя, вам может понадобиться «отправить» основание (прямоугольник корпуса) на задний план, чтобы заливка не прикрывала символы «+» и «-», которые, конечно, нужно поместить на передний план.

ГРАФИЧЕСКИЕ И ТЕКСТОВЫЕ СТИЛИ

ВВЕДЕНИЕ

ISIS реализует современную систему, которая позволяет вам управлять видом схем в терминах стиля линий, цвета заливки, шрифта, текстовых эффектов и т. д. Система достаточно мощная и позволяет вам управлять некоторыми или всеми аспектами вида схем глобально, но позволяя некоторым объектам нести их собственные локальные атрибуты.

Все графические объекты в ISIS (изображения компонентов, соединения, точки соединения и т. д.) отрисовываются соответственно графическим стилям. Графический стиль полностью определяет то, как будет нарисован и залит графический образ (такой как линия, прямоугольник, круг или что-то ещё), и включает атрибуты для стиля линии (сплошная, прерывистая, пунктирная и т. п.), толщины, цвета, стиля заливки, цвет фона и переднего плана и т. д. Аналогично, все этикетки и текстовые блоки в ISIS (названия портов, имена выводов и т. п.) рисуются согласно со стилем текста. Стиль текста включает описание того, как выглядит текст и включает атрибуты для вида шрифта (то есть, Arial, Times Roman и т. д.), высоты букв, их ширины, цвета и т. п.

В ISIS большинство объектов, как графика, провода, названия портов и т. д., каждый имеет свой локальный стиль, но подразумевается, что они могут меняться на основе объект-за-объектом; например, один провод может отличаться по виду от другого. Термин «локальный» используется для обозначения того, что установки стиля локальны для объекта. Другие объекты, как имена выводов, обозначения подсхем и т. п., всегда рисуются в одном из предустановленных стилей, отчего эти объекты можно менять только в плане «все-или-ничего». Например, подсхема может выглядеть так, как вам нужно, но так будут выглядеть все подсхемы.

Большинство объектов при размещении, имеющих собственный стиль, имеют локальный стиль, создаваемый из наиболее распространённых глобальных стилей. Например, когда вы помещаете порт, его стиль названия автоматически выбирается как TERMINAL LABEL стиль, а когда вы проводите соединение, его графический стиль автоматически определяется как WIRE стиль. Плоские графические объекты (2D Graphics Objects) в этом слегка отличаются, для этих объектов в *окне выбора* отображается список доступных графических стилей, а вновь размещённые графические объекты имеют вид в текущем выбранном стиле.

Теперь перейдём к «заумностям». Каждый локальный стиль сохраняет след глобального стиля, используемого для его инициализации. Вдобавок, каждый локальный стиль также имеет набор *Follow Global* опций при выборе, показывающих, что атрибут, ассоциированный со стилем, всегда принимает значение глобального стиля и, когда выбран, это значение локального стиля атрибута будет использовано. По умолчанию, для всех вновь размещаемых объектов, все опции *Follow Global?* выбираются такими, что вид вновь размещаемых объектов полностью следует глобальному стилю, на котором он основан.

Выигрыш от применения локального и глобального стилей, и возможность локальному стилю следовать некоторым или всем атрибутам глобального, таков:

LABCENTER ELECTRONICS

- это позволяет вам редактировать полный вид проекта через единственное редактирование глобального стиля — вам не нужно редактировать все объекты индивидуально;
- это позволяет вам определить библиотеку символов, которая автоматически «вмешивается» в вид рисунка, на котором они помещаются;
- это позволяет вам «зафиксировать» некоторые или все элементы вида компонента или другого объекта.

Например, представьте, что вы создаёте новый компонент и добавляете его в библиотеку. Предлагаемый вам рисунок графики компонента в COMPONENT стиле, впоследствии, когда компонент будет помещён на чертёж, автоматически примет COMPONENT стиль для этого чертежа.

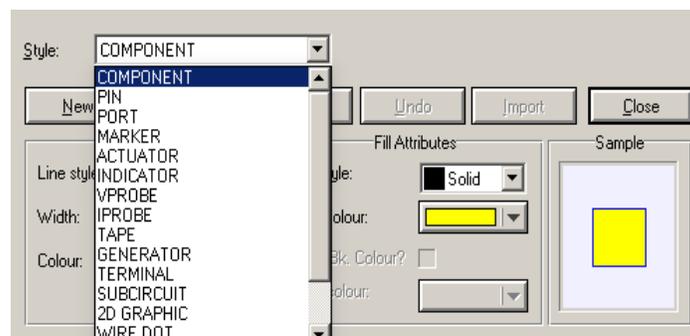
РУКОВОДСТВО

В качестве точки отсчёта для руководства по графическим и текстовым стилям, загрузите пример проекта STYLETUT.DSN из директории «Samples\Tutorials» вашей инсталляции Proteus. Как вы можете видеть, базовый цвет схемы в этом проекте — синий контур с жёлтой заливкой для компонентов и красный для портов. А если вы увеличите какой-либо текст, вы увидите, что он выполнен векторным Labcenter Electronics шрифтом.

Редактирование глобальных стилей

Посмотрим, как редактируются глобальные стили. Эти стили, как свидетельствует их имя, глобальны для всего проекта и их редактирование позволяет вам сделать глобальные изменения в том, как будет выглядеть схема.

Давайте, начнём с изменения цвета компонентов в схеме. Все компоненты в проекте были добавлены из стандартных библиотек, и, как следствие, графика компонентов вся выполнена в стиле COMPONENT. Мы изменим этот стиль, используя команду *Set Graphics Styles* из раздела *Template* основного меню. Выберите эту команду и вы увидите диалоговое окно *Edit Global Graphics Styles*. Этот диалог даёт доступ ко всем графическим стилям, определённым для текущего проекта — нажмите на кнопку выбора стиля (*Style:*), чтобы увидеть весь список.



Как показывают кнопки под списком стилей, возможно создавать, редактировать и удалять новые стили в дополнение к редактированию «предопределённых» стилей.

Убедитесь, что выбран стиль COMPONENT из списка *Styles*. Вы увидите атрибуты стиля в секциях *Line Attributes* и *Fill Attributes*. Под *Line Attributes* выберите красный цвет и под *Fill Attributes* выберите в *Fill Style*: атрибут «none». После этих изменений вы увидите, что в окошке предварительного вида форма обновится, показывая новый стиль.

Теперь выберите из списка стилей TERMINAL. Изменения, сделанные вами для стиля COMPONENT автоматически сохранились, а диалоговое окно изменилось, отображая установки для стиля TERMINAL. В этот раз под *Line Attributes* выберите синий цвет для линий и под *Fill Attributes* выберите *Fill Style*: «solid», что активизирует *Fg. Colour*:, где вы можете выбрать жёлтый цвет заливки.

Закройте диалоговое окно, используя кнопку **Close**, и схема будет перерисована, отображая изменения.

Новый цвет схемы, возможно, будет работать лучше на белом фоне, а поскольку вы, видимо, будете печатать проект на белой бумаге, давайте теперь выберем белый цвет, как наш «paper» цвет. Из раздела *Template* основного меню выберите команду *Set Design Defaults*. Диалог *Edit Design Defaults* позволит вам отредактировать большую часть стандартных цветов, используемых ISIS, включая цвет фона или «paper» цвет. Измените его на светло серый или белый и закройте диалог. Вы увидите, как изменились графические стили. А что стиль текста? Его изменение такое же, как изменение стилей текста, исключая, что стиль атрибутов, который вы получите, будет разным.

Однако прежде, чем начинать изменять стили текста для индивидуальных элементов, давайте изменим шрифт, используемый в чертеже «en mass». Как вы уже заметили, на чертеже в настоящее время весь текст выполнен векторным Labcenter Electronics шрифтом. Чтобы изменить предустановленный шрифт проекта, вновь выберите команду *Set Design Defaults* из раздела *Template* основного меню, что вызовет появление диалога *Edit Design Defaults*. Под *Font Face For Default Font* вы увидите, что выбран Vector Font — измените это на стандартный Windows Times New Roman и закройте диалоговое окно. Проект перерисовывается и весь текст будет отображён шрифтом Times New Roman.

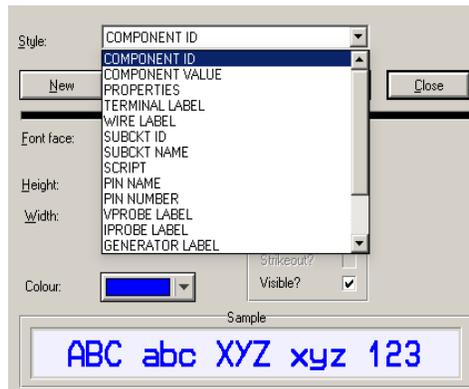
Где бы ISIS ни предлагал список TrueType™ шрифтов (как часть редактирования стиля текста), везде есть два дополнительных вида шрифта, находящихся в верхней части списка. Это Default Font и Vector Font. Опция *Default Font* размещает поддержку для любого выбранного шрифта в диалог *Edit Design Defaults*, а опция *Vector Font* выбирает внутренний Labcenter Electronics Vector шрифт. В случае STYLETUT.DSN весь текст и названия в проекте были помещены и/или отредактированы с использованием шрифта Default Font, что означает — все отображено в наборе шрифта в диалоге *Edit Design Defaults*; выберите здесь новый шрифт, и весь текст в проекте изменится немедленно!

Основное применение Vector Font: те случаи, когда проект выводится на перьевой плоттер. Windows не поддерживает правильный вывод шрифтов TrueType на плоттеры, а Labcenter Plotter Driver выводит весь текст шрифтом Vector Font, независимо от того, как это определено на чертеже. Vector Font имеет дополнительное преимущество — он гарантирует вывод точно того размера шрифта на любой жёсткой копии, что и виден на экране. А это не всегда получается с TrueType шрифтами.

LABCENTER ELECTRONICS

С другой стороны, если вы будете использовать только «bitmap» устройства печати, то есть, принтеры, эта особенность вам не нужна, и вы можете пользоваться всеми вашими TrueType™ шрифтами, как вы считаете целесообразным. Но нужно сказать, что схема, использующая более двух шрифтов, выглядит довольно «своеобразно»!

Чтобы использовать в нашем проекте в качестве «default font» шрифт Times New Roman, давайте сделаем некоторые специальные изменения в стиле текста объекта. Из *Template* основного меню выберем команду *Set Text Styles*, открывающую диалог *Edit Global Text Styles*. Заметьте, как похожа эта диалоговая форма на *Edit Global Graphics Styles*, которую вы видели раньше — как мы говорили, редактирование стилей текста очень похоже на редактирование графических стилей.



По умолчанию, COMPONENT ID будет выбран в окне *Style:*, а в *Font Face:* определено Default Font, что, как мы объясняли, отражает поддержку для шрифта, выбранного в диалоге *Edit Design Defaults*, и в настоящее время — это Times New Roman. По этой причине некоторые поля на форме диалога не подходят для TrueType™ шрифтов, как, например, ширина букв, и они остаются серыми. Измените поле *Font Face* на Courier New и выберите *Bold?* в опциях *Effects*. Теперь выберите стиль PIN NUMBER в списке *Style:* (изменения для COMPONENT стиля автоматически сохраняются), а затем снимите флажок *Visible?* в опциях *Effects*. Используйте кнопку **Close**, чтобы выйти из диалога.

Вы увидите, что на перерисованной схеме component ID (идентификаторы компонентов) написаны жирным шрифтом Courier New, и что номера выводов операционного усилителя больше не видны. Последний эффект порой полезен, когда чертёж предназначен для иллюстрации или блок-диаграммы.

Редактирование локальных стилей

Пока мы меняли только «глобальные» стили. Сделанные изменения сказывались на всей схеме. Теперь посмотрим, как изменить «локальные» стили на примере редактирования этикетки U1 операционного усилителя.

Начнём с обзора локального стиля: выделите этикетку U1 (и операционный усилитель) правым щелчком мышки и щёлкните по *Edit Label* в выпадающем меню, чтобы открыть диалоговое окно. Выберите закладку *Style*, на которой все установки «локального» стиля. Вы увидите

следующее: *Global Style*: селектор сверху, набор атрибутов стиля (*Font face*, *Width* и т. д.) слева, и для каждого атрибута флажки *Follow Global?* справа. В этом формате и текстовые, и графические локальные стили. Где бы ни стоял флажок *Follow Global?*, соответствующий атрибут стиля будет взят из глобального стиля, выбранного в селекторе *Global Style*. Снимая флажок *Follow Global?*, вы открываете возможность диалогового управления для связанного атрибута стиля, позволяющее вам задать локальное (специфическое для объекта) значение этого атрибута. Не все флажки *Follow Global?* доступны, как в этом случае, атрибут *Width* не имеет смысла для TrueType™ font face.



Изменения, которые мы собираемся внести, касаются изменения цвета этикетки U1. Начнём со снятия флажка *Follow Global?* справа от управления *Colour*:, что активизирует окно, в котором отображается текущая цветовая установка, с которой локальный стиль следует глобальному (в данном случае стилю COMPONENT ID). Измените цвет на темно синий, затем закройте диалог и снимите выделение с операционного усилителя, что перерисует его и он примет свои нормальные цвета. Этикетка U1 стала темно синей.

Чтобы проверить, что цвет этикетки теперь действительно не зависит от глобального стиля, тогда как остальные атрибуты все ещё следуют глобальному стилю, выберите команду *Set Text Styles* в разделе *Template* вновь, измените цвет COMPONENT ID стиля (выбранного по умолчанию) на пурпурный (magenta), установите флажок опции *Italic* и закройте диалог. Этикетка операционного усилителя остаётся темно синей, поскольку её цвет не следует больше глобальному COMPONENT ID стилю, хотя его атрибут курсив (*Italic*) все ещё следует глобальному стилю, он стал курсивом. Все остальные components ID (идентификаторы компонента) остались под влиянием глобального COMPONENT ID стиля, изменив цвет и отобразившись курсивом.

Для полноты картины мы сейчас покажем вам, как редактировать локальный графический стиль. Выберите иконку прямоугольника. Убедитесь, что выбран стиль COMPONENT, а затем, поместив курсор в верхний левый угол схемы, щёлкните левой клавишей и протащите прямоугольник вокруг (так, чтобы накрыть) схемы. Нам нужно изменить локальный стиль прямоугольника, сделав его прозрачным. Выделите прямоугольник, щёлкнув правой клавишей

LABCENTER ELECTRONICS

мышки по краю, выберите команду *Edit Properties* из выпадающего меню, чтобы попасть в диалоговое окно *Edit Box's Graphic Style*. Отрешившись от различных атрибутов стиля, заметьте как эта форма похожа на диалог с закладкой *Style* выше. Операции с этой формой такие же, как и с диалогом *Edit Label*. Снимите флажок *Follow Global?* с управления *Fill Style*, а затем, теперь оно отключено, измените стиль на «none». Закройте диалоговую форму, нажав на кнопку **ОК**, и снимите выделение с прямоугольника.

Это простой пример гибкости в управлении локальными и глобальными стилями, хотя в общем случае использования ISIS вам вряд ли понадобится модифицировать локальный стиль. Основное использование локальных стилей с плоскими графическими объектами (линии, прямоугольники и т. п.) требуется при создании новых библиотечных элементов, но даже и тогда мы рекомендуем избегать установок локальных стилей, пока этого можно избежать. Как вы могли видеть, как только объект получает установки локального стиля, они сохраняются, пока не будут переустановлены, а, когда они применяются к элементам библиотеки, это означает, что некоторые из них или все могут не гармонировать со схемой, в которую они вставляются. При правильном использовании, это обычно осознанно, получится, например, что основание транзистора всегда будет иметь сплошную заливку, невзирая на заливку, выбранную в глобальной установке для графики COMPONENT стиля. Однако при неправильном использовании, это может перейти на элементы библиотеки, которые просто не подходят по цвету к другим схемам или разным пользователям.

Работа с шаблонами (Template)

В заключение мы коснёмся такого предмета, как шаблоны. Вся информация о стилях текста, графики, виде точек пересечения и размере, цвете бумаги и т. п. — фактически все вы можете менять с помощью команд раздела *Template* основного меню; он называется шаблоны проекта. После инсталляции ISIS работает с predetermined шаблоном, который хранится в файле DEFAULT.DTF в директории библиотеки вашей системы Proteus. Когда ISIS стартует, или когда вы выбираете команду *New*, ISIS загружает начальные установки шаблона из этого файла, а все изменения, сделанные для шаблона, возвращаются на хранение в этот файл, если вы выбираете команду *Save Default Template* из раздела *Template*. Существующий шаблон не резервируется, так что не используйте команду *Save Default Template*, пока не будете полностью уверены, что вы очень хотите использовать эти установки шаблона в будущем (например, вы хотите всегда иметь свой «домашний» стиль оформления проекта).

Для портативности и устойчивости ISIS всегда сохраняет копию текущего шаблона в файлах схемы проекта и переписывает его при загрузке файла. Вы уже видели это при загрузке учебных проектов и, если вы позже загрузите некоторые из других примеров, полученных с ISIS, вы увидите, что они демонстрируют широкое разнообразие стилей.

Итак, что случится, если вы установите «домашний» стиль, используя команду *Save Default Template*, чтобы сделать его predetermined шаблоном, создадите проект и сохраните множество схем, а потом решите изменить «домашний» стиль? Простейший ответ — использовать команды *Apply Template From Design* или *Apply Default Template* из раздела *Template* основного меню. Обе эти команды загрузят новый шаблон в текущий проект либо из файла другого проекта, либо из файла predetermined шаблона, хотя заметьте, что объекты

с установками локального стиля будут продолжать пользоваться ими. Таким образом, вы можете установить новый предопределённый шаблон, а затем применить его ко всем вашим старым проектам. В качестве альтернативы, если вы потеряли ваш шаблон по умолчанию, вы можете восстановить его из старого проекта, а затем сохранить его, используя команду *Save Default Template*.

ШАБЛОНЫ И МЕНЮ *TEMPLATE*

Вся информация, управляющая видом схемы — стили графики, текста, цвета проекта, размер точек соединения и форма, и т. п. — называется шаблон (*template*). Все команды модификации шаблона находятся в меню *Template* и последовательно описаны ниже. Заметьте, что любые изменения шаблона сказываются только на текущей, запущенной копии ISIS, хотя они и сохраняются, и зарезервируются при сохранении любой схемы проекта. Чтобы сделать изменения доступными в последующем, при следующем начале нового проекта, вам нужно использовать команду *Save Default Template* из раздела *Template* основного меню, что обновит предопределённый шаблон.

ВВЕДЕНИЕ

ISIS широко использует концепцию свойств. Свойства задуманы для соответствия ключевых слов, идентифицирующих отдельные свойства, и значений, которые придаются этим свойствам для отдельных объектов. Например, в согласии с ARES, мы используем свойство, названное PACKAGE, и которое содержит контактные площадки для разводки печатной платы (PCB footprint).

Свойства могут быть связаны с объектами, листами и, собственно, проектом, а отношения между этими разными типами нуждаются в ясном понимании, если вы намерены взять лучшее из экстраординарной и мощной системы, не похожей на другие схемотехнические программы, которые мы видели.

СВОЙСТВА ОБЪЕКТОВ

Есть два отдельных типа свойств объекта — системные и пользовательские свойства. Разработчик включил набор зарезервированных ключевых слов, которые имеют специальные функции внутри самой программы ISIS, тогда как позже они могут относиться либо к внешним программам, как ARES и ProSPICE, либо могут относиться к вашему собственному, особому использованию программы.

Системные свойства

Системные свойства — это свойства, чьи ключевые слова имеют специальное значение внутри ISIS. Например, свойство DEVICE объекта компонент непосредственно определяет элемент библиотеки, связанный с ним. Некоторые из этих свойств текстовые, то есть, этикетки компонента REF и VALUE прямо доступны из диалога *Edit Component*, но другие, такие как вышеупомянутое свойство DEVICE регулируются как результат графических операций.

В основном вам нужно позаботиться только о системных свойствах, если вы хотите прочитать их значения командами поиска и выделения, или модифицировать их с помощью *Property Assignment Tool* (инструмент назначения свойств). Например, вы можете захотеть выделить все компоненты 7400 в вашем проекте. Это потребует от вас знания того, что системное свойство, содержащее имя элемента библиотеки, называется DEVICE.



Детально системные свойства, используемые для каждого типа объекта описаны в главе «Спецификация объектов».

Пользовательские свойства

Компоненты, подсхемы и гаджеты VSM могут нести неограниченное количество добавочных свойств в дополнение к их стандартным системным свойствам. Эти пользовательские свойства содержатся в текстовом блоке, известном как блок свойств и состоящих из строк вида:

```
SUPPLIER=XYZ Electronics
```

Вы можете редактировать эти блоки пользовательских свойств непосредственно в диалоговых формах разных объектов, либо манипулировать ими с помощью Property Assignment Tool.

Чтобы редактировать пользовательские свойства объектов:

1. Вызовите диалог свойств объекта двойным щелчком по нему левой клавишей мышки.
2. Если объект может поддерживать пользовательские свойства, диалог будет иметь окно текстового редактора, озаглавленное Properties. Поместите курсор ниже уже существующего текста и щёлкните левой клавишей мышки.
3. Отредактируйте текст, как нужно. Каждое из свойств должно состоять из одной строки, содержащей ключевое слово и значение, разделённые знаком равенства (=).

Ключевое слово пользовательских свойств будет, как правило, состоять только из букв, чисел и подчёркиваний. Ни в коем случае не должно быть пробелов, запятых, двойных кавычек или знаков равенства (, "=").

Согласно с основным поведением текста в ISIS, если заданное свойство заключено в фигурные скобки, («{» и «}»), тогда оно не будет отображаться на экране. Например, ввод:

```
{PRIMITIVE=DIGITAL}
```

определит объект, как требующий цифровой симуляции, но этот текст не будет отображаться.

Иногда нужно, чтобы появлялось только значение, в этом случае можно сделать так:

```
{MODFILE=}OPAMP
```

Теоретически вы можете помещать фигурные скобки в других местах. Однако, когда Property Assignment Tool изменяет блок свойств, он работает в предположении, что если используется вообще, то фигурные скобки были размещены, как показано в примерах выше. Если же они были размещены иначе, тогда вы можете получить неожиданные результаты.

Определение свойств (PROPDEFS)

Есть возможность поддержки дополнительных деталей при спецификации пользовательских свойств устройства. Например, наиболее общими пользовательскими свойствами компонента будут PACKAGE и MODFILE. Задание подходящих Property Definitions (определение свойств) было создано для библиотек компонентов; эти свойства отображаются в собственных полях диалоговой формы *Edit Component*. Определение свойств включает описание свойств, тип данных (таких как целое, с плавающей точкой или строковое) и для числовых данных правильный диапазон входных величин. Предопределённое значение может также быть (и часто

так и есть) задано.

Подобная схема существенно облегчает понимание того, какие свойства подойдут для конкретной модели, и что эти свойства означают. Также появляется возможность добавить альтернативные типы упаковки и модели для симулятора путём, очевидным для конечных пользователей разных элементов библиотеки.

За дополнительной информацией о том, как создавать и работать с определением свойств для ваших собственных элементов библиотеки, вы можете обратиться к разделу «Определение свойств» далее.

Свойства, которые не известны для конкретного устройства (или все свойства устройства, которое не имеет определённых свойств), тем не менее появляются в текстовом блоке свойств, как описано выше, а схема, за счёт этого, имеет обратную совместимость со старыми проектами и/или элементами библиотек, не имеющих определения свойств.

СВОЙСТВА ЛИСТА

Введение

Каждый лист схемы может поддерживать набор заданных свойств. Это может быть задумано как определение констант (числовых или текстовых), используемых для назначения внутренних свойств объекта на отдельном листе. Сами по себе, они могут быть не столь важны, но их реальный смысл становится актуален, когда они используются в контексте выражений для свойств объекта.

Например, если свойства листа определены в блоке так:

```
*DEFINE  
PI=3.142
```

вы сможете затем определить значение резистора выражением:

```
VAL=EVAL(500/PI)
```

При создании спецификации netlist подобный синтаксис заставит компилятор netlist'a рассчитать свойство и значение резистора появится там или в спецификации материалов (bill of materials) как 159.134.



Больше информации о выражениях свойств можно найти в разделе «Вычисление выражений свойств».

Определение свойств листа

Свойства листа можно определить следующим образом:

- Непосредственно, используя текстовый блок DEFINE. Вы можете использовать такое определение для использования в выражениях, как в примере выше.
- Как результат параметрического картирования, используя текстовый блок MAP ON. В

этом случае значение исходного свойства, заданное с помощью заявления MAP ON, используется для выбора набора определений свойств листа из таблицы. Это наиболее часто используется при создании универсальных моделей симуляции, в которых некоторые схожие устройства моделируются с использованием той же схемы, но с разными наборами свойств листа для каждого из типов устройств.

Дальнейшее обсуждение этого можно найти в руководстве к Proteus VSM.

- Наследованием свойств родительского объекта. Другими словами, если родительский объект имеет заданное свойство:

R3=10k

то производный лист автоматически получит его в качестве свойств листа. Этим создаётся базис для параметризованных цепей (раздел «Параметризованные цепи»), в которых несколько образцов заданного иерархического модуля могут получать разные значения компонента.

Если те же свойства определены на производном листе в блоках DEFINE или MAP ON и в родительском блоке свойств объекта, будет использовано значение из родительского блока свойств. Смысл таков: дать predetermined значения для свойств листа, которые могут быть отменены, если потребуется.

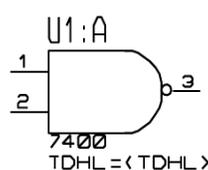
Правила области для свойств листа

Очень важно сделать так, чтобы свойства листа могли относиться только к листу, для которого они определены. В частности, свойства родительского листа не доступны никакому производному, пока они не пройдут через родительский блок свойств объекта. Если вам не нужен такого рода доступ к свойствам, вы можете добавить строку вида:

TDHL=<TDHL>

для соответствующего родительского объекта(ов). Если TDHL определено как свойство листа на родительском листе, оно станет свойством объекта на родительском объекте, и, таким образом, будет определено как свойство листа для производного листа, где оно может, в свою очередь, появиться в добавочных выражениях свойств объекта.

РОДИТЕЛЬСКИЙ ЛИСТ

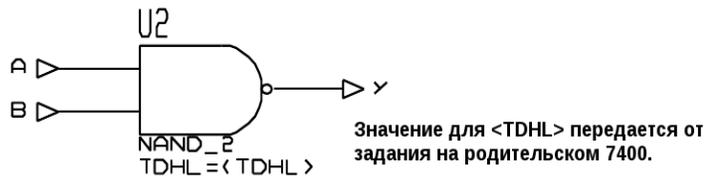


*DEFINE
TDHL=10n

Значение для <TDHL> здесь передается от свойств листа, определенных в блоке *DEFINE.

Такой механизм аналогичен передаче параметров в компьютерной программе на языке Си.

ПРОИЗВОДНЫЙ ЛИСТ



СВОЙСТВА ПРОЕКТА

Свойства проекта для данной схемы определяются аккумулярованием всех свойств листа, декларируемых на основном листе проекта. Поскольку основные листы не могут иметь родительских, из этого следует, что свойства проекта могут декларироваться только использованием текстового блока DEFINE на первом листе.

Для вывода netlist'a SDF свойства проекта появляются в блоке PROPERTIES netlist'a, и могут быть интерпретированы любым приложением, прочитывающим netlist. В случае Proteus VSM свойства проекта используются для определения опций симулятора, таких как количество шагов, рабочая температура и т. п. Специфика этого дана в руководстве к Proteus VSM.

Для создания списка свойств проекта:

1. Перейдите на первый (root) лист вашего проекта с помощью команды *Goto Sheet* в разделе *Design* основного меню.
2. Выберите иконку *Script* на инструментальной панели *Mode Selector*.
3. Переместите курсор туда, где вы хотели бы видеть блок определения свойств и щёлкните левой клавишей мышки.
4. Введите на первой строке текста
*DEFINE
5. Введите нужные назначения свойств.

Заметьте, что свойства проекта также и свойства листа, на котором они определены. Однако стандартные правила области для свойств листа все ещё применима, и свойства проекта не доступны для выражений свойств на других листах.

ПАРАМЕТРИЗОВАННЫЕ ЦЕПИ

Введение

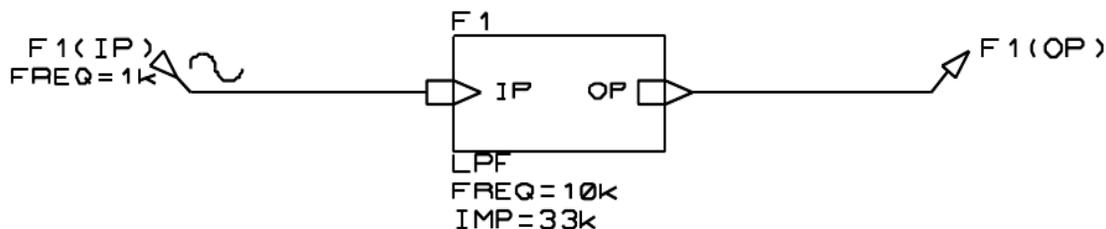
ISIS имеет уникальные и достаточно мощные возможности для комбинации свойств листа, свойств объекта и иерархической разработки в применении к параметризованным электрическим цепям. Параметризованные цепи таковы, что некоторые значения компонента (или другие свойства) задаются в виде формулы, а не постоянной величины. Естественно, формула содержит переменные или параметры, а значения для них берутся из определения свойств листа для отдельных образцов цепи. И потом, следуя этому в контексте иерархического проекта, разные образцы цепи могут иметь разные параметры, а, следовательно, разные значения компонентов.



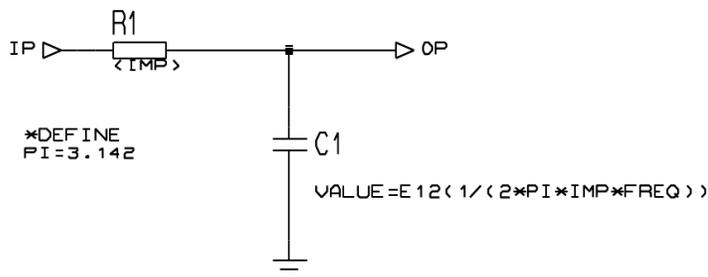
Больше информации об иерархических проектах можно найти в главе «Иерархические проекты» далее. Если вы не представляете, что такое иерархический проект, мы предлагаем вам пропустить этот раздел до лучших времён!

Пример

Параметризованные цепи лучше всего проиллюстрировать на примере проекта, который вы найдёте под именем файла LPF.DSN в директории «Samples» после установки Proteus. Основной (root sheet) лист этого проекта показан ниже:



Проект состоит из единственной подсхемы, которая имеет пользовательские свойства, определённые для требуемой частоты и импеданса фильтра. Модуль, прикрепленный к подсхеме, это фактическая параметризованная цепь:



LABCENTER ELECTRONICS

Есть несколько моментов, на которых следует остановиться:

- Блок DEFINE определяет одно свойство листа: PI. И оно играет роль константы в выражении свойств для значения конденсатора.
- Поле значения резистора содержит строку <IMP>. Этот синтаксис с угловыми скобками (<>) приводит к тому, что компилятор netlist'a заменяет <IMP> на значение свойства листа IMP, которое равно 33 кОм. В этом случае нет нужды вычислять выражение — замещение чисто буквенное.
- Конденсатор имеет присвоенное пользовательское свойство (в его блоке свойств), которое задаёт его значение через выражение. Функция E12 определяет и то, что выражение должно быть рассчитано компилятором netlist'a, и то, что результат должен быть округлён до ближайшего E12 значения. Другие опции — EVAL (не округлено) и E24 (округлено до E24 значения).

PI, IMP и FREQ — это все свойства листа. PI появляется из блока DEFINE, тогда как IMP и FREQ приходят из родительской подсхемы.

Если вы сгенерируете Bill of Materials, вы увидите следующее:

```
QTY  PART-REFS          VALUE
---  -
Resistors
-----
1    R1                33k

Capacitors
-----
1    C1                470p
```

ISIS вычислил, что $1/(2*3.142*33000*10000)$ даёт приблизительно 0.000000000482, и затем округлил это до ближайшего значения E12 — 470 пФ.

Фактически, в приведённом примере имели место два отдельных процесса: замещение свойств и вычисление выражения свойств. Оба имеют свои преимущества и свои недостатки, и в следующем разделе они обсуждаются подробнее.

Замещение свойств

Это механизм, который используются для значения резистора, и будет производиться, когда бы компилятор netlist'a не встретил значение свойства, содержащее ключевое слово свойства, заключённое в угловые скобки (<>). Если ключевое слово совпадает со свойством листа, тогда выражение в угловых скобках замещается значением свойства листа. Если нет свойства листа, тогда генерируется предупреждение netlist'a, а свойство удаляется из объекта.

Есть два случая, где замещение свойства полезно:

- Вы можете использовать параметризованные цепи, в которых параметры не числовые. Корпуса для разводки плат — это, пожалуй, самый общий такой пример; все прекрасно, ISIS рассчитал, что используется конденсатор в 470 пФ, но вам ещё, возможно, нужен и корпус для разводки печатной платы. Если прикрепить к конденсатору пользовательское свойство:

PACKAGE=<C1_PACKAGE>

тогда вы можете добавить свойство:

C1_PACKAGE=CAP10

в подсхему. Когда создаётся netlist, C1 появится со свойством PACKAGE=CAP10. Вычисление выражения не подходит для этой цели, поскольку CAP10 не вычисляется, как число.

- Другое, более важное использование замещения свойства — это задание для анализа качания параметра (sweep analyses) в Proteus VSM. В этом вы хотите, чтобы симулятор вычислил выражения, не ISIS, и тогда может быть уместнее создать свойства компонента, используя замещение свойств, вместо вычисления выражения свойств. Дальнейшее обсуждение этого вы найдете в руководстве по VSM.

Вычисление выражения свойства

В отличие от замещения свойств, которое является чисто текстовым процессом, вычисление выражения свойств вовлекает ISIS в числовые расчёты значения свойства, имеющего вид формулы, с последующей заменой её значением. Дополнительно ISIS может также округлять полученное число к E12 или E24.

Есть три формы синтаксиса:

EVAL (...)

E12 (...)

E24 (...)

Во всех случаях скобки должны содержать математическое выражение, которое может включать: операции сложения (+), вычитания (-), умножения (*), деления (/) и значения. Значения могут быть постоянными или именами свойств листа. Умножение и деление имеют наивысший приоритет, но можно использовать скобки, чтобы изменить это, как требуется.

Ниже показаны некоторые примеры выражений:

EVAL (1 / (A+B))

A и B - это свойства листа.

E12 (20k+2 *F*PI)

20k автоматически трактуется как 20000.

E24 (3+4 *5)

Вычисление до 24.

Хотя в некоторых случаях это более мощный вариант, чем замещение свойств, но есть

LABCENTER ELECTRONICS

ограничения:

- Расчёты могут поддерживать только числовые значения — выражения, содержащие строки, не допускаются.
- Вы можете использовать только ссылочные свойства листа в формуле — вы не можете получить другие свойства объекта или значения других компонентов.
- Нет поддержки математических функций (как синус, косинус, квадратный корень, и т.д.).

Некоторые из этих трудностей мы, возможно, устраним в следующих версиях.

Функции округления E12 (), E24 ()

Механизм расчёта выражений свойств поддерживает возможность округления результата до ближайшего E12 или E24 значения. Это предупреждает параметризованные цепи от завершения работы с недействительными или многозначными значениями с плавающей точкой.

Вы должны иметь в виду:

- Округление будет невозможно для выражений, которые приводят к нулевому или отрицательному значению. Маловероятно, что это станет проблемой, поскольку отрицательным значениям резисторов и конденсаторов неоткуда взяться.
- Округление выполняется на геометрической, а не арифметической основе, так что середина между 3k3 и 4k7 получается приблизительно 3.94. Это, мы чувствуем, остаётся за пониманием, собственно, E12 и E24.
- Не забывайте, что если параметризованная цепь содержит несколько округлённых значений, нет механизма, позволяющего округлять все в оптимальном направлении, принимая во внимание, что они могут взаимодействовать. Это означает, что для критически важных разработок фильтров и тому подобного, вам, может быть, лучше рассчитать все вручную (с учётом различных вариантов сочетаний), и использовать замещение параметров для загрузки выбранных вами значений непосредственно в схему.

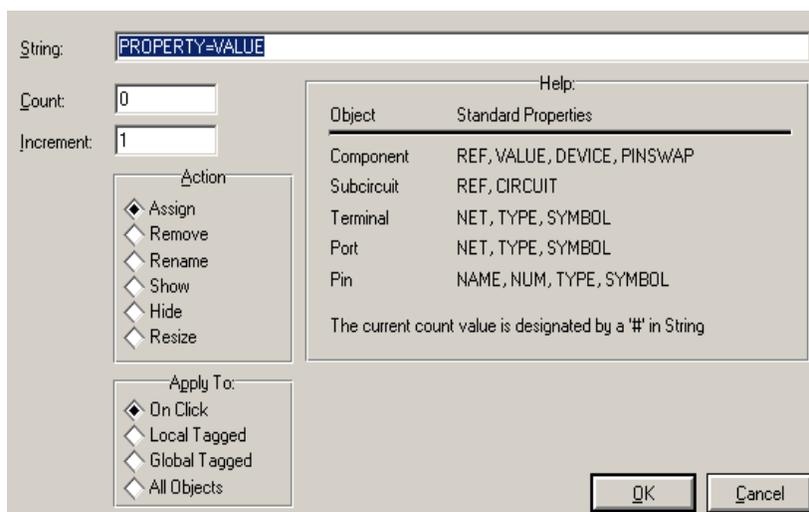
Конечно, если вы используете Proteus VSM, вы можете запустить симуляцию и увидеть, как значения ISIS сказываются на работе цепи.

PROPERTY ASSIGNMENT TOOL

Property Assignment Tool (PAT, инструмент присвоения свойств) — это инструмент, который позволяет вам присваивать, удалять, показывать и скрывать свойства объекта, принадлежащие выделенному объекту вашего чертежа.

Диалоговое окно PAT

PAT оперирует с не самой сложной формой диалога, имеющей следующие поля:



String

Назначение свойства или ключевое слово свойства, которым выбранное действие будет выполняться для каждого объекта.

Count

Начальное значение для счётчика. Счётчик увеличивается на единицу для каждого вызова PAT. Текущее значение счётчика может быть включено в строку вводом символа «решётки» (#).

Action

Действие, которое вы хотите выполнить. В разделе «Действия PAT» есть описание различных типов действий.

Apply

Режим, в котором вы хотели бы, чтобы PAT оперировал. Различные типы работы описаны в разделе «Режимы работы PAT».

Действия PAT

Property Assignment Tool может выполнять следующие действия:

ASSIGN

Строка может содержать назначение свойства в форме:

keyword=value

и это свойство будет назначено выбранному объекту.

Если вы хотите задать значения, которые будут следовать друг за другом, например, D0, D1, D2 и т. д., тогда используйте символ решётки (#) в значении и установите начальное значение счётчика, как нужно.

И пользовательские, и системные свойства могут присваиваться; присвоение системных свойств могут вызывать изменение графики вашего чертежа.

REMOVE

Строка должна содержать только ключевое слово свойства и это свойство будет удалено у выбранного объекта. Удалять можно только пользовательские свойства.

RENAME

Строка может содержать оба присвоения свойства в форме:

`current_keyword=new_keyword`

Имя слева от знака присваивания — это существующее имя свойства, которое вы хотите переименовать; имя справа — это новое имя, которое вы хотите присвоить.

Переименовываться могут только пользовательские свойства.

SHOW

Строка должна содержать только ключевое слово свойства и это свойство будет сделано видимым для выбранного объекта.

Могут быть показаны и пользовательские, и системные свойства.

HIDE

Строка должна содержать только ключевое слово свойства и это свойство будет сделано невидимым для выбранного объекта.

Могут быть скрыты и пользовательские, и системные свойства.

RESIZE

Строка должна содержать присвоение, как

`REF=20,16`

и заданное свойство будет присвоено высоте и ширине выбранного объекта. Только текстовые системные свойства могут менять размер.



Изменение размера текста этикетки полностью задаст атрибуты Height и Width стиля текста этикетки как локальные, соответственно, изменения глобального стиля текста для этикетки не будет отражаться на этих этикетках.

Режимы работы РАТ

Выбранное действие РАТ может быть применено к проекту следующими путями:

ON CLICK

При выборе кнопки **OK**, выбирается иконка *Instant Edit*. Каждый объект, по которому вы щёлкните левой клавишей мышки, претерпит заданное действие.

Когда вы выбираете другую иконку, РАТ отключается.

Этот режим — единственный способ применить РАТ к соединениям, в порядке назначения названия проводам. ISIS не может выполнить РАТ присвоение на выделенных проводах, поскольку он не знает, где размещать этикетки.

LOCAL TAGGED

Выбранное действие выполняется для всех выделенных объектов, но только текущего листа.

Вы можете выделить объекты либо индивидуально, либо с помощью команды *Search & Tag*.

GLOBAL TAGGED

Выбранное действие выполняется для всех выделенных объектов по всему проекту.

Вы можете выделить объекты либо индивидуально, либо с помощью команды *Search & Tag*.

ISIS способен запомнить разные состояния выделения для разных образцов объектов в иерархическом проекте, если это появляется. Однако блоки пользовательских свойств обобщаются между образцами, так что вы не можете изменить пользовательские свойства только для одного образца.

ALL OBJECTS

Выбранное действие выполняется для всех объектов проекта.

Команда *Search and Tag*

Команда *Search and Tag* облегчает выбор отдельных групп объектов для последовательного процесса РАТ опций *Local Tagged* или *Global Tagged*.

Есть три команды поиска:

SEARCH

Это обычная операция поиска, которую вы используете, чтобы начать поиск. Она выделяет объекты, которые встречает по условию поиска и очищает выделение объектов, которые не подходят.

AND SEARCH

Эта операция поиска может быть использована для исключения объектов из текущего набора выделения. Она очищает выделение всех объектов, которые не удовлетворяют условиям поиска, и оставляет те, что удовлетворяют.

OR SEARCH

Эта операция поиска используется для включения объектов в уже выделенный набор. Она выделяет все объекты, которые удовлетворяют условию поиска, и оставляет без выделения те, что не удовлетворяют.

Примеры

Команды *Property Assignment Tool* и *Search & Tag* — это мощные и гибкие инструменты применимые для работы со свойствами объектов. Однако они могут напугать начинающих. С учетом этого, пошаговые примеры должны дать вам начальное представление.

Чтобы маркировать ответвления шины:

1. Вызовите РАТ нажав клавишу «А».
2. Задайте строку NET=D# и щёлкните **ОК**. Действие определится как *Assign* и режим *On Click*.
3. Щёлкайте левой клавишей мышки по каждому проводу, где хотите видеть этикетку. Курсорные клавиши и клавиша **Enter** могут использоваться в качестве альтернативы мышке. Соединения, по которым вы щёлкните, получают новые этикетки в последовательности D0, D1, D2 и т.д.

Чтобы присвоить корпуса для всех BC108 в проекте:

1. Вызовите команду *Search and Tag*, нажав клавишу «Т».
2. Задайте свойство VALUE и строку BC108 и щёлкните по **ОК**. Режим будет задан как *Equals*. Все компоненты со значением BC108 будут выделены.
3. Вызовите РАТ клавишей «А».
4. Задайте строку PACKAGE=TO18 и щёлкните **ОК**. Действие будет определено как *Assign*, а режим как *Global Tagged* (подразумевая, что выделены BC108). Все выделенные BC108 получают новое пользовательское свойство.

Для переименования всех ИТЕМ свойств в свойство CODE:

1. Вызовите РАТ клавишей «А».
2. Задайте строку ИТЕМ=CODE, действие *Replace* и режим *All Objects*, затем щёлкните **ОК**. Все объекты со свойством ИТЕМ=value будут переименованы со свойством CODE=value.

Чтобы скрыть все свойства упаковки:

1. Вызовите РАТ клавишей «А».
2. Задайте строку PACKAGE, действие *Hide* и режим *All Objects*, затем щёлкните **ОК**. Все свойства PACKAGE будут скрыты.

Чтобы изменить размер обозначения компонента:

1. Вызовите ПАТ клавишей «А».
2. Задайте строку REF=10,8, действие *Resize* и режим *All Objects*, затем щёлкните **ОК**. Все обозначения компонентов будут уменьшены до нового размера.

Чтобы присвоить большие корпуса конденсаторам 1000 мкФ:

1. Вызовите команду *Search and Tag*, нажав клавишу «Т».
2. Задайте свойство DEVICE и строку CAP ELEC, затем щёлкните **ОК**. Этим выделятся все электролитические конденсаторы.
3. Вызовите команду *AND Search* из меню *Tools*.
4. Установите свойство VALUE, строку 1000u и режим *Begins*, затем щёлкните **ОК**. Это коснётся конденсаторов со значением 1000u или 1000uF.
5. Вызовите ПАТ клавишей «А».
6. Задайте строку PACKAGE=ELEC-RAD30 и щёлкните **ОК**. Действие будет определено как *Assign*, а режим как *Global Tagged* (если были выделены конденсаторы). Все выделенные конденсаторы получают новые корпуса.

ОПРЕДЕЛЕНИЕ СВОЙСТВ

Создание *Property Definitions* (определение свойств)

Определение свойств вводится на странице *Component Properties* в диалоге *Make Device*. Более детально это описано в контекстно чувствительном help диалога и/или в инструкциях о том, как создавать устройства.

Предопределённые определения свойств

Некоторые свойства уже будут назначены для большинства устройств, которые вы создаёте. Например, все, что отправится в программу разводки печатной платы будет нуждаться в свойстве PACKAGE (корпус), а все, что имеет модель для симулятора в MODFILE, MODEL или SPICEMODEL свойствах. Вы также можете захотеть придать свои свойства, такие как STOCKCODE (складской код), SUPPLIER (поставщик) или COST (цена) для большинства устройств, которые создаёте.

Чтобы облегчить вам работу, есть список предопределённых свойств, который можно определить, используя команду *Set Property Definitions* из раздела *System* основного меню. Свойства, определённые этой командой, доступны в поле *Name* диалогового окна *Edit Device Properties*. Информация, управляемая этой командой, содержится в файле PROPDEFS.INI, находящемся в директории библиотеки вашей установки Proteus.

Старые проекты

Заметьте, что компоненты и элементы библиотеки в проекте, открытые в версии PROTEUS до 4.5 не будут содержать определения свойств. Если вы хотите, вы можете заставить ISIS придать предопределённые определения свойств этим компонентам «на лету», то есть, когда они редактируются.

Чтобы использовать такую возможность, вызовите команду *Set Property Definitions* из меню *System* и установите флажок *Apply Default Properties to Components in Old Designs*.

КОМПОНЕНТЫ

Компонент — это образец устройства, выбранный из одной из библиотек устройств. Поскольку некоторые из устройств много-элементные, получается, что в некоторых случаях, некоторые компоненты на схеме могут, фактически, все принадлежать к одному физическому компоненту на печатной плате. В этих случаях логические компоненты обозначаются как U1:A, U1:B, U1:C, U1:D, чтобы обозначить, что они принадлежат одной микросхеме. Эта форма обозначения также позволяет ISIS выбрать правильный набор номеров выводов для каждого элемента.

Не считая физических выводов, компоненты — это только объекты, которые вы можете размещать и которые будут создавать физические сущности при разводке печатной платы. Все, что ещё вы можете разместить, используется либо для задания соединений, либо только для декоративных целей. Это важный момент, поскольку попытка использовать другие ISIS объекты (в частности, графические символы) для представления объектов на печатной плате будет обречена на провал.

Выбор компонентов из библиотек устройств

Когда вы запускаете ISIS с пустым листом, *навигатор выбранных объектов* тоже пуст. Прежде, чем вы сможете размещать какие-либо компоненты, вы должны их выбрать из библиотек в *навигатор*.

Чтобы выбрать компоненты из библиотек устройств:

1. Убедитесь, что в *навигаторе выбранных объектов* будут показаны устройства, выбрав иконку (под всеми иконками появляются подсказки при наведении курсора) *Component Mode* на инструментальной панели выбора режимов.
2. Щёлкните левой клавишей мышки по кнопке **P** в окне *навигатора*. Это вызовет появление диалогового окна *Pick Devices* с категориями компонентов.
3. В окне обзора *категорий объектов (Category:)* выберите категорию, в библиотеке которой вы хотите указать устройство.
4. Если есть подкатегории объектов, то они появятся в окне *Sub-category:*, выбор в котором ограничит количество объектов библиотеки. То же относится к выбору в окне производителей *Manufacturer*.
5. Выберите нужное устройство и щёлкните по кнопке **ОК**, что приведёт к появлению объекта в *навигаторе выбранных объектов*.



Диалоговое окно *Pick Devices* можно масштабировать, изменяя размер диалога до требуемого. Для вызова диалога можно использовать щелчок правой клавишей мышки в рабочем поле, в выпадающем меню: *Place-Component-From Library*.



Больше информации о библиотеках можно найти в разделе «Возможности»

библиотеки».

Размещение компонентов

Размещение компонентов следует общей схеме, описанной в разделе «Размещение объектов».

Чтобы разместить компонент:

1. Если устройства нужного типа нет в окне навигации выбранных объектов, вначале выберите нужный объект из библиотеки, как описано выше.
2. Подсветите, щёлкнув левой клавишей мышки, имя устройства в *окне навигации*. ISIS покажет предварительный вид устройства в *окне обзора*.
3. Используйте иконки Rotation и Mirror для придания устройству нужного положения.
4. Щёлкните левой клавишей мышки по свободному месту в окне редактора, контур устройства, в который превратился курсор, позволяет вам перетаскивать устройство, уточняя место расположения. После повторного щелчка устройство окажется на чертеже.

Замещение компонентов

Поскольку при удалении компонента удаляются провода, подходящие к нему, изменение компонента его удалением, а затем размещением нового компонента может оказаться утомительным. Вместо этого ISIS поддерживает специальный механизм, облегчающий такого рода операции.

Для замещения компонента:

1. Щёлкните по новому типу устройства, чтобы подсветить его в *окне навигатора*, как описано выше.
2. Используйте иконки Rotation и Mirror для придания нужного положения новому устройству, соответственно тому, как вы хотите его видеть на чертеже.
3. Позиционируйте курсор мышки внутри старого компонента так, чтобы хотя бы один из его выводов совпал с выводом старого компонента. Щёлкните левой клавишей мышки и выберите **ОК** в сообщении о замене компонента.

ISIS постарается заменить старый компонент новым, сохраняя столько соединений, сколько сможет. Программа ищет совпадений по положению, а затем по имени вывода. Попытка замещения слишком разных компонентов, похоже, не даст положительного результата, но тогда используйте команду Undo, чтобы вернуться в исходное положение, если попытка замены оказывается явно неудачной.



Также возможно осуществить замену компонента через свойства DEVICE, используя *Property Assignment Tool*. Смотрите раздел «Свойства» выше.

Редактирование компонентов

Компонент можно редактировать, используя все основные методы редактирования (см. «Редактирование объектов»), а также специальные с использованием команды *Edit Component* из раздела *Edit* основного меню (горячая клавиша «Е»). Как и в большинстве механизмов редактирования в PROTEUS, появится диалоговое окно с нужными полями для редактирования. Однако в диалоге *Edit Component* доступные поля зависят от заданных свойств компонента. Вы можете обнаружить, что этот диалог может резко отличаться для разных компонентов. Контекстно чувствительный Help доступен для диалоговой формы, но, пожалуйста, заметьте, что, по причинам указанным выше, некоторые поля могут не иметь подсказок, связанных с ними.

Вы можете поправить появление компонентов, выводы, имена выводов и т. д., редактируя связанную с ними глобальную графику и стиль текста. Смотрите раздел «Редактирование глобальных стилей».

Свойства компонентов

Компоненты имеют следующие системные свойства:

Имя свойства	Описание
REF	Этикетка указателя ссылки компонента.
VAL	Этикетка значения компонента.
VALUE	Этикетка значения компонента или пользовательское свойство VALUE, если заданный текст слишком велик для этикетки.
DEVICE	Элемент библиотеки. Если задан, это вызовет автоматическую логику замещения, которая постарается сохранить соединения.

Если другие имена свойств (как STOCKCODE, TOLERANCE и т. д.) имеют место, тогда будут созданы пользовательские свойства в текстовом блоке компонента.

Скрытые выводы питания

Если компонент был определён со скрытыми выводами питания, тогда шина, к которой эти выводы будут присоединены, может стать видима или отредактирована щелчком по кнопке **Hidden Pins** в диалоге *Edit Component*.

По умолчанию скрытые выводы присоединяются к шине с этим же именем — например, скрытый вывод VDD соединится с VDD, а скрытый вывод VSS, с VSS и т.д.

ТОЧКИ

Точки пересечения используются для отметки соединений между проводами. В основном ISIS размещает и удаляет их автоматически, но иногда полезно помещать точку в заданную позицию

и затем соединять провода через неё.

Соединения, которые касаются или пересекают провода, никогда не присоединяются, пока не будет точки в месте контакта. И наоборот, везде, где есть точка, всегда будет присоединение, пока вы накладываете провода и точки, перемещая их в контакте друг с другом.



Вы можете поправить появление соединительных точек, редактируя графический стиль WIRE DOT. Смотрите раздел «Редактирование глобальных стилей». Также вы можете установить их размер и форму, используя команду *Set Junction Dots*.

Размещение точек

Для размещения точки:

1. Выберите иконку *Junction Dot Mode* на панели выбора режимов.
2. Укажите, где вы хотите поместить точку в окне редактора.
3. Щёлкните левой клавишей мышки, чтобы поместить точку.

Автоматическое размещение точек

ISIS автоматически размещает точки, всякий раз, когда вы проводите соединения от существующего провода, так что здесь получается 3 точки в этом месте.

Автоматическое удаление точек

Когда провод или провода удаляются, ISIS обнаруживает, где остаются точки с двумя или без присоединённых проводов. Такие точки автоматически удаляются пока это не приводит к прерыванию замкнутой цепи.

МАРКЕРЫ ПРОВОДОВ

Маркеры проводов используются для задания отдельных имён соединениям, группам проводов и выводов, и также для задания свойств определённым соединениям. Это, фактически, не присуще объектам в общей схеме программы. Вместо этого, их поведение похоже на другие этикетки, такие как названия и значения компонентов. Из чего следует, что процедуры по их размещению и удалению несколько отличаются от тех, что осуществляются для других объектов.



Смотрите «Генерация спецификации (Netlist)» о роли и этикетках соединений в спецификации и связности.

Размещение и редактирование этикеток проводов

Чтобы разместить или отредактировать этикетку провода:

1. Выберите иконку *Wire Label* на панели выбора режима.
2. Если задавать новую этикетку, поместите курсор над проводом в позиции, где вы хотите видеть маркировку или, для существующей этикетки, точку где-то по ходу провода или на самой этикетке.
3. Щёлкните левой клавишей мышки для размещения этикетки. Появится диалоговое окно *Create Wire Label* (создать этикетку провода) или *Edit Wire Label* (редактировать этикетку провода).



Смотрите раздел «Редактирование локальных стилей», где найдёте детали по управлению на закладке *Style*.

4. Введите требуемый текст для маркировки провода.
5. Щёлкните по **ОК** или нажмите **ENTER**, чтобы закрыть диалог.

Заметьте следующее:

- Вы не можете размещать этикетку на проводе иную, чем уже есть на проводе.
- Вы можете поместить более, чем одну этикетку на провод. Если вы хотите, чтобы они все имели одинаковое имя, и чтобы все они обновлялись автоматически при изменении любого из имён, установите флажок *Auto-Sync*.
- ISIS будет ориентировать этикетку согласно ориентации сегмента провода, на котором она размещается. Это можно изменить в диалоге *Edit Wire Label*.

Чтобы изменить вид соединения:

1. Убедитесь, что иконка *Wire Label* не выбрана.
2. Выделите провод двойным щелчком по нему. Появится диалоговое окно *Edit Wire Style*.
3. Снимите флажок *Follow Global?* того атрибута графического стиля, который вы хотите изменить. Если атрибут стиля и флажок *Follow Global?* не активны, это означает, что атрибут стиля незначителен и задаётся другими установками атрибута, либо локальными, либо всегда следует глобальному стилю.
4. Задайте требуемые установки стиля атрибута.
5. Нажмите **ENTER** или щёлкните по кнопке **ОК**, чтобы закрыть диалоговое окно и сохранить изменения. Нажмите клавишу **ESC** или щёлкните по кнопке **CANCEL**, чтобы закрыть диалог и отказаться от изменений.



Чтобы изменить вид всех проводов в схеме, используйте команду *Set Graphics Style* в разделе *Template* основного меню для редактирования графического стиля WIRE.



См. раздел «Графические и текстовые стили», где больше информации о стилях.

Удаление маркировки проводов

Чтобы удалить этикетку провода:

1. Выделите этикетку провода двойным щелчком по ней.
2. Появится диалоговое окно *Edit Wire Label*. Заметьте, что совместное выделение провода и этикетки приведёт к появлению диалога *Edit Wire Style*. Чтобы открыть диалог *Edit Wire Label*, следует выделить только этикетку.
3. Убедитесь, что строка текста маркировки полностью выделена (так будет по умолчанию) и нажмите клавишу **DEL**, чтобы удалить её.
4. Щёлкните по **ОК** или нажмите **ENTER**, чтобы закрыть диалог и сохранить изменения.

Использование маркировки провода для доступа к имени сети

Обычное применение маркировки провода — это обозначение отдельного соединения, и все выводы, к которым он графически подходит, будут принадлежать данной сети. Тогда эта группа выводов считается присоединённой к любой другой группе выводов, которая тоже имеет такое же имя сети, даже если графическое соединение отсутствует. Иногда кто-то может маркировать отдельные сети чисто для лучшего визуального восприятия.



См. раздел «Имена сетей», где информации больше.

Использование этикеток проводов для доступа к свойствам сети

Свойства сети используются для задания специальной информации о сети. В системе PROTEUS основное использование этой возможности — задание стратегии разводки при использовании ARES. Назначение свойств сети имеет форму:

```
<prop>=<value>
```

Например, этикетка провода со строкой

```
STRAT=POWER
```

приведёт к тому, что свойство STRAT для соединения получит значение POWER. В контексте ISIS/ARES это приведёт к тому, что ARES использует POWER стратегию для создания соединений.



См. «Свойства сети и стратегия разводки», где найдёте больше информации о стратегиях разводки.

Свойства этикетки провода

Провод или этикетка провода (они одинаковы для этой цели) имеют следующее свойство:

Имя свойства	Описание
NET	Текст маркировки провода

Этикетки проводов могут быть заданы только через *Property Assignment Tool*, когда инструмент используется в режиме *On Click*. Это связано с тем, что у ISIS нет возможности определить, где должна быть размещена маркировка провода на выделенном соединении, пока вы не укажете положение этикетки мышкой.

SCRIPTS

Основное достоинство ISIS в его возможности поддерживать текстовые скрипты свободного формата и многообразное их применение. Это включает:

- Определение переменных для использования в выражениях свойств и карты параметров.
- Определение моделей примитивов и скриптов для использования с симулятором ProSPICE.
- Аннотирование проекта со значительным объёмом текста.
- Сохранение информации о свойствах и упаковке, когда компонент раскладывается на составные части (decomposed).

Размещение и редактирование скриптов

Процедуры размещения и редактирования скриптов почти идентичны, поскольку обе операции вызывают диалог *Edit Script Block*.

Чтобы разместить скрипт:

1. Выберите иконку *Text Script Mode* на панели выбора режима.
2. В окне редактора укажите место, где вы хотите видеть верхний левый угол блока и щёлкните левой клавишей мышки. Появится диалог *Edit Script Block*.
3. Введите текст нового скрипта в поле *Text*. Вы можете также поменять атрибуты скрипта в данный момент — посмотрите «Чтобы отредактировать скрипт» ниже, где детально описывается форма диалога.
4. Закройте диалог с помощью кнопки **ОК** для размещения текста или нажмите **CANCEL** для выхода из диалога и отказа от размещения скрипта.

Чтобы отредактировать скрипт:

1. Либо: (a) Выделите скрипт двойным щелчком левой клавиши мышки. (b) Выделите скрипт и нажмите **CTRL+E**, чтобы отредактировать его.

Появится диалоговое окно *Edit Script Block*.

2. Если требуется, измените атрибуты скрипта.

Диалог *Edit Script Block* имеет две закладки: *Script* и *Style*. Загляните в руководство по графическим и текстовым стилям («Редактирование локальных стилей»), где больше информации о редактировании локального текстового стиля.

3. Закройте диалог.

Чтобы закрыть диалоговое окно и сохранить изменения либо щёлкните по **ОК**, либо используйте **CTRL+ENTER** клавиши. Чтобы закрыть диалог и отказаться от изменений либо щёлкните по кнопке **CANCEL**, либо используйте клавишу **ESC** на клавиатуре.



Вы можете изменить размер диалогового окна *Edit Script Block*, чтобы сделать больше поле *Text*. Вы можете сохранить новый размер, используя левую кнопку на панели заголовка, где выберите команду «Изменить размер».

Типы блоков скриптов

ISIS в настоящее время поддерживает следующие типы блоков скриптов:

ТИП БЛОКА СКРИПТА	ЗАГОЛОВОК БЛОКА
Назначение свойств элемента	*FIELD
Назначение глобальных свойств сети листа	*NETPROP
Определение свойств листа	*DEFINE
Таблица картирования параметров	*MAP ON varname
Таблица определения моделей	*MODELS
Именованные скрипты	*SCRIPT type name
Скрипты SPICE модели	*SPICE type name

Краткое описание использования и формат каждого типа дано в разделах ниже.

Назначение свойств элемента (***FIELD**)

Поля блоков изначально предназначены для задания свойств разъемов по их физическим выводам. Особый смысл это приобретает тогда, когда требуется использовать физические соединители, а на схеме нет целого элемента, представляющего разъем, к которому свойства могут быть непосредственно отнесены.

Пример блока показан ниже:

*FIELD

J1,PACKAGE=CONN-D9

J2,PACKAGE=CONN-D25

Этим присваивается J1 для PCB корпуса 9 выводного D разъёма и J2 для 25 выводного корпуса соединителя.

Вы можете использовать это для назначения свойств обычному компоненту, но смысла в этом мало, поскольку их можно применить непосредственно к компоненту.



См. «Физические соединители», где о них есть больше информации.



См. «Isis и Ares» далее, где обсуждаются корпуса для программы разводки.

Назначения глобального свойства сети листу (*NETPROP)

Свойство цепи (net) обычно назначается размещением на соединении этикетки с синтаксисом:

```
prop=value
```

Однако есть случаи, когда вам хотелось бы придать одинаковые свойства большому количеству цепей. Чаще всего, когда всем цепям отдельного листа нужно задать специфический тип (возможно, сделать очень широкой дорожку для питания) и для этого можно использовать блок NETPROP. Например, если блок:

```
*NETPROP
```

```
STRAT=POWER
```

включён в лист, тогда все цепи, провода которых явно не внесены в свойства сети, получат свойство сети STRAT=POWER.



См. «Использование этикеток соединений для назначения свойств цепи».

Определение свойства листа (*DEFINE)

Свойство листа в сущности — это переменная, которая определяется для данного листа и может быть использована в выражениях свойств внутри списков свойств компонентов на этом листе. Свойства листа, определённые на первом листе проекта, также отражаются в спецификации (netlist) и могут быть использованы, как параметры управления такими программами, как VSM симуляторы, которые читают netlist.

Пример блока показан ниже:

```
*DEFINE
```

```
TEMP=40
```

```
MINSTEPS=100
```

Этим определяется два свойства TEMP и MINSTEPS, и они будут помещены на первый лист проекта, с тем чтобы они передавались в VSM, как параметры управления симуляцией.



См. раздел «Свойства». Там более подробно обсуждается управление свойствами.

Таблицы картирования параметров (*MAP ON varname)

Это задумано в качестве весьма перспективной возможности, и для тех, кому очень важно участие в создании универсальных моделей, используемых с VSM. Дальнейшую информацию можно найти в руководстве VSM, но для полноты картины мы предлагаем пример:

```
*MAP ON VALUE
7400
: TDLH=12n,TDHL=7n
74LS00 : TDLH=10n,TDHL=6n
74S00 : TDLH=5n,TDHL=3n
```

Эта таблица будет помещена на схеме модели для 2х-входового вентиля НЕ-И, и будет выбирать разные значения для TDLH и TDHL, согласно со значением (то есть, типом) родительского объекта. Также производные листы будут использовать NAND_2 примитивы со свойствами:

```
TDLH=<TDLH>
TDHL=<TDHL>
```

Во время работы netlist Parameter Mapping Tables будут проверять свойство родительского объекта VALUE, и если, скажем, это был тип 74LS00, тогда свойства листа TDLH=10n и TDHL=6n будут определены для этих образцов модели. Затем, при обработке NAND_2 примитива, эти значения будут замещены <TDLH> и <TDHL>, так что эти примитивы приобретут правильные времена, свойственные модели 74LS00.

Случай DEFAULT теперь поддержан.

Таблицы определения модели (*MODELS)

Эти блоки используются только с VSM симуляторами (в настоящем, фактически, только с аналоговыми симуляторами) и предоставляют метод условного обозначения для обслуживания большого числа свойств, которые используются некоторыми примитивами симулятора. Например, модель биполярного транзистора имеет около 30 разных свойств, и будет довольно бесполезно, если вы получите доступ ко всем этим индивидуальным свойствам для каждого из транзисторов в схеме.

Вместо этого вы можете использовать блок MODELS для определения свойств отдельных типов транзисторов, и затем ссылаться на весь набор свойств, используя имя модели.

Пример блока показан ниже:

LABCENTER ELECTRONICS

```
*MODELS
741_NPN : BETA=80,ISAT=1E-14,RB=100,VA=50,\
          TAUF=0.3E-9,TAUR=6E-9, CJE=3E-12,CJC=2E-12
741_PNP : BETA=10,ISAT=1E-14,RB=20, VA=50,\
          TAUF=1E-9, TAUR=20E-9,CJE=6E-12,CJC=4E-12
```

Здесь определены два типа транзисторов, используемых в микросхеме 741. Индивидуальные транзисторы могут затем приобретать значения 741_NPN или 741_PNP, а VSM автоматически будет передавать им характеристики, определенные в таблице модели.



См. руководство к VSM с обсуждением свойств моделей примитивов.

Именованные скрипты (*SCRIPT scripttype scriptname)

В некоторых случаях полезно иметь возможность экспорта именованного блока текста во внешнее приложение. В частности, SPICE и SPICE-AGE форматтеры netlist'a используют именованные скрипты PSPICE и SPICE-AGE для поставки информации управления для симуляторов.

Типичный именованный блок скрипта показан ниже. Заметьте, что конец текста отмечен ключевым словом ENDSCRIPT, хотя это опция, если нет других блоков в объекте скрипта.

```
*SCRIPT PROGRAM 7493
// Объявление привязки к выводам и локальным переменным:
PIN SCA, СКВ, RA, RB
PIN QA,QB,QC,QD
INT counta = 0, countb = 0

// Поддержка одно-битового 'A' счётчика присвоение выхода:
IF RA=H THEN
counta = 0
ELIF SCA=HL THEN
counta = (counta+1) % 2
ENDIF
QA = counta & 1

// Поддержка трех-битового 'B' счётчика и присвоение выхода:
IF RB=H THEN
countb = 0
ELIF СКВ=HL THEN
countb = (countb+1) % 8
ENDIF
QB = countb & 1
QC = countb & 2
QD = countb & 4
*ENDSCRIPT
```



См. примечания к приложениям SPICE и SPICE-AGE о синтаксисе именованных скриптов, связанных с этими симуляторами.

Скипт модели SPICE (*SPICE)

Эти скрипты позволяют входным картам (input cards) SPICE печататься на схеме и загружаться непосредственно в ProSPICE во время симуляции. Это может быть удобно, если вы хотите разработать или протестировать SPICE модели в исходном SPICE формате.



За дальнейшей информацией обратитесь к руководству Proteus VSM.

ШИНЫ

Шина — это разновидность условного обозначения большого числа проводов, и обычно используется в микропроцессорных схемах. ISIS имеет беспрецедентный уровень поддержки шин, включая не только возможность провести шины между иерархическими модулями, но также и для определения элементов библиотеки с выводами шины. Следовательно, возможно соединить CPU с массивом памяти и периферией единственной шиной вместо упорядочивания пучка проводов, вводов шины и шин.

Размещение шин

Шины размещаются, как правило, тем же путём, что и отдельные провода, исключая то, что они должны идти «к» и «от» точек соединения шины, а не точек соединения провода. Также, в отличие от проводов, шины могут размещаться изолировано от других объектов.

Чтобы разместить шину:

1. Выберите иконку *Buses Mode* на панели выбора режимов.
2. Укажите место, где вы хотели бы видеть начало шины. Это может быть вывод шины, уже существующей шины, или свободное место на чертеже.
3. Щёлкните левой клавишей мышки, чтобы начать шину, затем щёлкните ещё раз для каждого из углов на выбранном пути.
4. Чтобы закончить шину на точке соединения с шиной (вывод шины или существующая шина), укажите на неё и щёлкните левой клавишей мышки. Чтобы закончить шину на свободном месте, щёлкните левой клавишей мышки.

Этикетки шин

Шина может быть маркирована совершенно также, как и соединение. Однако ISIS определяет специальный синтаксис для этикеток шины.

Чтобы разместить или отредактировать этикетку шины:

1. Выберите иконку *Wire Label Mode* на панели выбора режимов.
2. Укажите, где вы на шине хотели бы разместить маркировку.
3. Щёлкните левой клавишей мышки для размещения этикетки. Появится диалоговое окно

Create Wire Label или *Edit Wire Label*.



См. «Редактирование локальных стилей».

4. Введите требуемый текст для этикетки соединения. Что-нибудь похожее на D[0..7] или A[8..15]. Если вы пропустите задание диапазона, тогда шина примет значение ноль для базы и ширину от самой широкой шины, присоединённой к ней. В общем, вам лучше всегда задавать диапазон.
5. Щёлкните по **ОК** или нажмите **ENTER**.

Заметьте следующее:

- Вы не можете размещать маркировку отличную от этикетки соединения или шины.
- Вы не можете размещать более одной этикетки на одной выделенной шине. Попытка это сделать изменит существующую этикетку.
- ISIS будет ориентировать этикетку согласно ориентации сегмента соединения, на котором она размещается. Это можно изменить в диалоге *Edit Wire Label*.

Для удаления этикетки шины:

1. Выделите соединение и этикетку указав на неё.
2. Щёлкните правой клавишей мышки по этикетке и из выпадающего меню выберите *Delete Label*.

Для изменения вида шины:

1. Убедитесь, что иконка Wire Label Mode не выбрана.
2. Выделите шину, указав на неё, и щёлкните правой клавишей мышки, выберите из выпадающего меню *Edit Wire Style*.

Появится диалог *Edit Wire Style*.

3. Снимите флажок *Follow Global?* того атрибута стиля графики, который хотите изменить. Если атрибут стиля и его флажок оба не активны, это значит, что атрибут не важен и определён другими установками стиля, либо локальными, либо через следование глобальному атрибуту.
4. Задайте атрибут стиля.
5. Нажмите **ENTER** или щёлкните по **ОК**, чтобы закрыть диалог и сохранить изменения. Нажмите **ESC** или щёлкните по кнопке **CANCEL**, чтобы закрыть диалог и отказаться от изменений.



Чтобы изменить вид всех шин на схеме, используйте команду *Set Graphics Style* в разделе *Template*, чтобы отредактировать графический стиль BUS WIRE.

Соединение проводов/шин

Иногда, как раз при подготовке выводов шины, бывает необходимо отвести один сигнал от шины, возможно, для адресных целей. С чисто графической точки зрения, это просто повод разместить на шине провод.

Чтобы разместить отвод на шине:

1. Если вы планируете провести соединение от шины к другому объекту, убедитесь, что иконка *Buses Mode* не выбрана.
2. Создайте соединение обычным образом. Шина будет вести себя подобно обычному проводу в этой обстановке.

Может быть, вам будет полезно понять, что когда вы создаёте соединение таким путём, работает автоматическое создание точек соединения, чтобы дать что-то для провода и шины, где они могли бы соединиться. Однако вы не увидите точки, поскольку она приобретает тот же цвет, и тот же размер, что и шина.

После соединения провода с шиной вы должны задать, какой сигнал шины ответвляется в этом случае.

Чтобы маркировать ответвление шины:

1. Убедитесь, что на шине есть маркировка вида D[0..7]. Эта этикетка определяет восемь соединений, названных D0, D1, ... , D7.
2. Разместите этикетку на проводе, чтобы обозначить, какой из сигналов ответвляется.

Если шина не соединяется ни с выводами шины, ни с разъёмом, вы можете пропустить шаг [1]. В этих обстоятельствах шина сама не играет роли в определении соединений схемы.

Размещение отвода шины без маркировки будет отмечено, как ошибка netlist, так как это полностью двусмысленная ситуация. ISIS не может знать, какой сигнал вы пытаетесь вывести. Как, впрочем, и любой человек, читающий схему, не сможет этого сделать!

Свойства шины

Шина или этикетка шины (они эквивалентны в этом случае) имеют следующее свойство:

Имя свойства	Описание
NET	Текст этикетки шины

Этикетка шины может быть присвоена только с помощью *Property Assignment Tool*, когда инструмент используется в режиме *On Click*. Причина в том, что для ISIS нет способа определить, где будет размещена маркировка на выделенной шине, пока вы не укажете это с помощью мышки.

ПОДСХЕМЫ

Подсхемы используются для добавления листов нижнего уровня к листам верхнего уровня в иерархическом проекте. Каждая подсхема имеет имя, которое идентифицирует производный лист, и имя схемы, которое идентифицирует производную схему. На любом данном листе все подсхемы должны получить разные имена листов, но могут, и это часто так, иметь одинаковые имена схем. Больше информации об иерархических проектах вы найдёте в разделе «Иерархические проекты».

Подсхемы могут также иметь списки свойств, и это важно для параметризованных схем, в которых разные образцы данной электрической цепи могут иметь разные значения элементов (или другие свойства), как, впрочем, и независимое документирование. Больше информации об этом можно найти в разделе «Параметризованные цепи».

Размещение подсхем

Размещение подсхемы вызывает появление прямоугольника подсхемы, к которому позже можно добавить порты подсхемы. Одна и та же иконка используется для обеих операций; ISIS определяет, что происходит, соответственно тому, где курсор — на свободном месте или на существующем прямоугольнике подсхемы.

Чтобы разместить прямоугольник подсхемы:

1. Выберите иконку *Subcircuit Mode* на панели *выбора режима*.
2. Укажите, где бы вы хотели видеть левый верхний угол прямоугольника. Это должна быть точка, не занятая уже существующей подсхемой.
3. Щёлкните левой клавишей мышки и протащите курсор до нужного размера прямоугольника, затем щёлкните левой клавишей мышки еще раз.



Вы можете поправить вид подсхемы, редактируя *SUBCIRCUIT graphics style* (графический стиль подсхемы). См. «Редактирование глобальных стилей».

Чтобы разместить порты подсхемы:

1. Выберите иконку *Subcircuit Mode* на панели *выбора режима*.
2. Выберите тип порта, который вам нужен, в *окне выбора*.
3. Укажите, где вы хотите видеть порт. Это должна быть точка на левой или правой стороне подсхемы, к которой вы хотите присоединить порт.
4. Щёлкните левой клавишей мышки, чтобы поместить порт.

После создания порта или портов вы должны маркировать их. Иерархический проект работает через соединение портов родительского объекта с так же названными портами производного листа. Из этого следует, что и основные порты, и производные должны иметь уникальные имена. Это можно сделать разными способами:

- Редактируя этикетку порта производного листа методом, описанным в разделе «Редактирование объектов».
- Используя инструмент *Property Assignment Tool*, чтобы присвоить свойство NET одному или более портам.

Это особенно эффективно, если группа портов имеет имена, идущие по алфавиту.

Правильно будет, если вы присоедините шины к портам. В этом случае имя для порта должно обычно определяться диапазоном соединений шины, как, например, D[0..7], хотя это не обязательно. Если диапазон проводов не задан, ISIS будет использовать диапазон, заданный секции шины, которая присоединена к порту, или, если она не имеет имени или диапазона, тогда размер шины будет взят из выводов, соединённых с шиной.

Размещение порта без маркировки вызовет ошибку netlist, поскольку подобный объект не имеет смысла и не может быть связан с чем-либо на производном листе.

Редактирование подсхем

Можно изменить размер подсхемы, используя общую процедуру, описанную в разделе «Изменение размеров объекта», и редактировать, используя любую из техник редактирования, описанных в разделе «Редактирование объекта».



См. раздел «Параметризованные цепи», где больше информации о параметризованных цепях.

Свойства подсхем

Подсхемы имеют следующие свойства:

Имя свойства	Описание
NAME	Имя образца подсхемы. Также используется, как имя производного листа.
CIRCUIT	Имя производной схемы. Если вы не присвоите его, ISIS автоматически выберет его при первом входе на производный лист.

Если какие-либо другие свойства имён присвоены, тогда это создаст пользовательские свойства в текстовом блоке подсхемы. Такие свойства в последствии станут свойствами листа для производного листа и могут использоваться в выражениях свойств.

ВЫВОДЫ

Выводы (terminals) используются для задания интерфейса с электрической цепью — ISIS не допускает свободных проводов — оба конца должны присоединяться к чему-либо, так что все входы и выходы вашего проекта будут обозначены выводами.

LABCENTER ELECTRONICS

Есть два типа выводов — логические выводы и физические выводы (Logical Terminals и Physical Terminals). Оба различаются чисто синтаксисом своих этикеток.



Вы можете поправить вид выводов, редактируя *TERMINAL graphics style*. См. «Редактирование глобальных стилей».

Логические выводы

Логический вывод (terminal) служит просто заданием имени цепи для провода, к которому он присоединён. Группы проводов с одним (или больше) именем связываются вместе генератором netlist. Логические выводы, таким образом, поддерживают смысл соединения чего-либо вместе без использования проводов. В частности, они поддерживают соединение между листами в многолистовом проекте.

Как и с Wire Labels и Bus Entries (этикетки проводов и вводы шины), имена цепей могут содержать любые буквенно-цифровые символы плюс тире ('-') и подчёркивание ('_'). Пробелы могут использоваться внутри PROTEUS, но могут вызвать проблемы для других пограмм.

Логические выводы могут также соединяться с шинами. Так поддерживается весьма эффективный способ проводки шин вверх и вниз по иерархическому проекту.



См. раздел «Имена сетей» о роли логических выводов в netlist'ах.



См. «Иерархические проекты» о роли логических выводов в иерархических проектах.

Физические выводы

Физические выводы представляют выводы физических соединителей. Например, вывод с именем:

J3:2

означает вывод 2 разъёма J3. Пока их полностью можно определить по разъему, как и по другим компонентам (то есть, определить их представление устройством), использование физических выводов (Physical Terminals) имеет преимущество — выводы могут размещаться там, где удобнее.

При разводке печатной платы, где необходимо задать тип корпуса разъёма, должно быть использовано свойство FIELD блока присвоения (см. «Присвоение свойств элементу (*Field)»), поскольку нет реального компонента для редактирования.

Заметьте, что выводы шины могут не быть физическими, так как нет смысла задавать нумерацию выводов для индивидуальных выводов.

Размещение выводов

ISIS поддерживает неограниченное многообразие символов выводов. Однако, когда вы впервые

выбираете иконку *Terminal* (из выпадающего меню при щелчке правой клавишей мышки), базовый набор 7 типов автоматически загружается в *окно выбора*.

Чтобы поместить вывод:

1. Если тип вывода (terminal), который вам нужен, отсутствует в списке *окна выбора*, выберите его в библиотеке символов.
2. Подсветите имя вывода в *окне выбора*. ISIS покажет вид вывода в *окне обзора*.
3. Используйте иконки ориентации *Rotation* и *Mirror* согласно вашему желанию по размещению выводов.
4. Укажите место на краю объекта, где вы хотите, чтобы появился вывод, и щёлкните левой клавишей мышки.

Поместив вывод, вы должны обозначить его, поскольку без этого вывод будет игнорироваться компилятором netlist'a. Есть ряд подходов:

- Отредактируйте этикетку вывода, используя общие методы, описанные в разделе «Редактирование объекта».
- Используйте Property Assignment Tool для присвоения свойств NET одному (или больше) выводу. Это особенно эффективно, если группа выводов имеет имена, располагаемые по алфавиту или подряд.

Правильно будет, если вы присоедините шины к портам. В этом случае имя для порта должно обычно определяться диапазоном соединений шины, как, например, D[0..7], хотя это не обязательно. Если диапазон проводов не задан, ISIS будет использовать диапазон, заданный секции шины, которая присоединена к порту, или, если она не имеет имени или диапазона, тогда размер шины будет взят из выводов, соединённых с шиной.

Размещение порта без маркировки вызовет ошибку netlist, поскольку подобный объект не имеет смысла, логически или физически.

Редактирование выводов

Вывод можно редактировать, используя общие техники редактирования (см. раздел «Редактирование объекта»). Дополнительно, поскольку выводы часто появляются группами, Property Assignment Tool можно использовать для обозначений и задания электрического типа выводов. Диалог Edit Terminal имеет следующие поля:

Name Имя цепи для логического вывода или имя ножки (pin) для физического вывода.
Type Электрический тип вывода.

Свойства выводов

Выводы имеют следующие системные свойства:

Имя свойства	Описание
NET	Этикетка вывода сети.
SYMBOL	Символ, используемый для вывода. Это может быть один из стандартных символов вывода или имя определенного пользователем символа.
TYPE	Электрический тип вывода. Это может быть любой из PASSIVE, INPUT, OUTPUT, BIDIR или POWER.

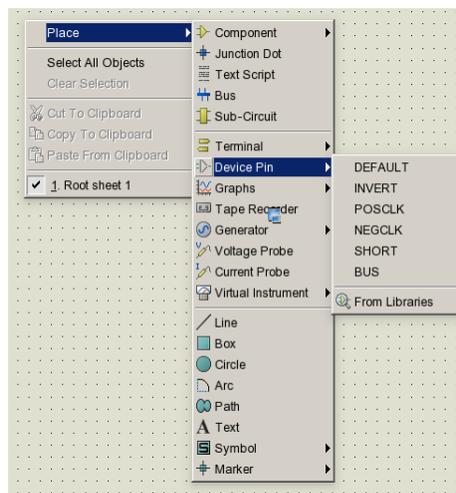
КОНТАКТЫ

Полное описание того, как создавать и редактировать ваши собственные устройства, есть в разделе «Возможности библиотеки». Здесь мы обсудим только размещение и редактирование таких объектов, как контакты (pin). Они используются для представления каждого вывода устройства на чертеже и состоят из нескольких графических элементов (часто из единственной линии) плюс возможность добавить и отобразить имя контакта и номер.

Пожалуйста, отметьте, что вы не можете проводить соединения вне контактов — вы можете только соединять выводы, которые принадлежат полностью оформленным компонентам, размещённым обычным образом.

Размещение контактов

Может быть определено неограниченное количество типов контактов, а правильный выбор предложен в SYSTEM.LIB. Однако при первом выборе *Device Pin* базовый набор 6 типов автоматически загружается в окно выбора, и они подходят для большинства целей.



Чтобы разместить контакт:

1. Если тип контакта, который вам нужен, отсутствует в списке *окна выбора*, попробуйте указать его в библиотеке символов.
2. Подсветите имя типа вывода в *окне выбора*. ISIS покажет вид вывода в *окне обзора*.
3. Используйте иконки *Rotation* и *Mirror* для ориентации контакта соответственно тому, как вы хотите разместить его.
4. Укажите положение в окне редактора, где вы хотели бы видеть вывод, и щёлкните левой клавишей мышки.

После размещения контакта вы, обычно, хотите модифицировать его — дать имя выводу, номер и электрический тип. Вот некоторые возможности:

- Редактируйте вывод вручную, используя общие методы, описанные в разделе «Редактирование объектов».
- Используйте *Property Assignment Tool*, чтобы присвоить свойства NAME, NUMBER и TYPE одному (или более) контакту. Это особенно эффективно, если группа выводов имеет имена, идущие по алфавиту или по порядку, как в шине.

Если вывод представляет шину данных или адреса, вы можете использовать вывод шины. В этом случае номер контакта может быть задан только с помощью *Visual Packaging Tool*. Равно и тогда, когда устройство имеет множество элементов (как 7400), вы должны вновь задать номера контактов для каждого элемента, используя *packaging tool*. Либо в этом случае вы оставляете номер вывода пустым.

Редактирование контактов

Контакты можно редактировать, используя любой из общих методов редактирования (см. «Редактирование объектов»). Дополнительно, поскольку выводы часто появляются в группе, хорошо работает *Property Assignment Tool* для определения имён, номеров и электрических типов.

Свойства контактов

Контакты имеют следующие системные свойства:

Имя свойства	Описание
NAME	Имя контакта.
NUM	Номер контакта.
SYMBOL	Символ, используемый для контакта. Это может быть стандартный символ или имя определённого пользователем символа.
TYPE	Электрический тип контакта. Это может быть любой из: PASSIVE, INPUT, OUTPUT, BIDIR, TRISTATE, PULLUP, PULLDOWN, POWER.

LABCENTER ELECTRONICS

ГАДЖЕТЫ СИМУЛЯТОРА

Интерфейс симулятора ProSPICE использует некоторые специальные объекты в ISIS. Полный набор состоит из:

GENERATORS **TAPES**
VOLTAGE PROBES **CURRENT PROBES**
GRAPHS

Инструкции относительно использования этих объектов даны в руководстве к Proteus VSM.

2D GRAPHICS (Графические элементы)

ISIS поддерживает следующие типы плоских графических объектов: lines (линии), boxes (прямоугольники), circles (круги), arcs (дуги), scalable text (масштабируемый текст) и composite symbols (смешанные символы). Они предназначены к использованию непосредственно на чертеже, например, для изображения разделительных линий и контуров выделения секций чертежа, а также для создания новых элементов библиотеки (устройств, символов, контактов и выводов — pins и terminals).

Размещение графических элементов

Далее описаны процедуры размещения разных типов графических объектов.

Чтобы поместить линию:

1. Выберите иконку *2D Graphics Line Mode* в окне выбора объекта.
2. Выберите *Graphics Style* для линии на чертеже из окна выбора.
3. Щёлкните левой клавишей мышки там, где будет начало линии.
4. Щёлкните ещё раз в конце линии.

Чтобы разместить прямоугольник:

1. Выберите иконку *2D Graphics Box Mode* в окне выбора объекта.
2. Выберите *Graphics Style* для прямоугольника на чертеже из окна выбора.
3. Укажите, где вы хотели бы видеть левый верхний угол элемента и щёлкните левой клавишей мышки.
4. Протащите курсор мышки до правого нижнего угла, где щёлкните еще раз.

Чтобы разместить круг:

1. Выберите *2D Graphics Circle Mode* в окне выбора объекта.

2. Выберите *Graphics Style* для круга на чертеже из *окна выбора*.
3. Переместите курсор в цент будущего круга и нажмите левую клавишу мышки.
4. Протащите мышку, определяя размер будущего круга и нажмите левую клавишу ещё раз.

Чтобы разместить дугу:

1. Выберите *2D Graphics Arc Mode* в *окне выбора объекта*.
2. Выберите *Graphics Style* для дуги на чертеже из *окна выбора*.
3. Продумайте вид дуги в одном из квадрантов эллипса — вы вначале определяете этот квадрант. Укажите курсором, где будет лежать один конец квадранта и нажмите левую клавишу мышки.
4. Протащите мышкой примерный участок квадранта и щёлкните ещё раз, когда нужный конец дуги будет определён.
5. Используйте жёлтые квадратики «поддержки» для придания нужного вида дуге.

Чтобы разместить произвольную фигуру (path):

1. Выберите *2D Graphics Closed Path Mode* в *окне выбора объекта*.
2. Выберите *Graphics Style* для дуги на чертеже из *окна выбора*.
3. Укажите мышкой место размещения первой вершины в *окне редактирования* и щелкните левой клавишей мышки.
4. Чтобы нарисовать прямолинейный участок, просто перемещайте курсор; чтобы нарисовать криволинейный участок, нажмите и удерживайте клавишу **CTRL** и перемещайте мышку.

При перемещении мышки отображается «резиновая» линия, показывая тип сегмента (прямой или нет), который будет создан в этом месте.

5. Щёлкните левой клавишей мышки в месте второй вершины. При размещении вершина не может быть ни удалена, ни убрана командой *Undo*, хотя фигура может редактироваться после размещения удалением ненужных вершин или изменением сегментов.
6. Повторите шаги четыре и пять для завершения фигуры и нажмите **ESC** для завершения ввода фигуры.

Фигура не завершена, пока вы не разместите последнюю вершину в той же точке, где разместили первую, чтобы закрыть контур.

Чтобы разместить графический текст:

1. Выберите *2D Graphics Graphics Text Mode* в окне выбора объекта.
2. Выберите *Graphics Style* для дуги на чертеже из окна выбора.
3. Используйте иконки *Rotation* и *Mirror* для придания нужной ориентации тексту на чертеже.
4. Укажите в окне редактирования место, где нижний левый край текста должен располагаться и щёлкните левой клавишей мышки.

Появится диалог *Edit 2D Graphics Text*.



См. «Редактирование плоской графики» относительно формы диалога.

5. Введите текст в форму диалога и установите выравнивание, размер шрифта и т.д., если нужно.
6. Нажмите **ENTER** или щёлкните кнопку **OK** для размещения текста, нажмите **ESC** или щёлкните **CANCEL**, чтобы прервать размещение текста.

Чтобы разместить символ:

1. Выберите *2D Graphics Symbols Mode* в окне выбора объекта.
2. Выберите символ из библиотеки символов. Вызвать диалог *Pick Symbols* можно щелчком по кнопке **P** в окне выбора объекта.
3. Используйте иконки *Rotation* и *Mirror* для ориентации символа перед размещением на чертеже.
4. Установите курсор в место, где следует поместить символ в окне редактора и щёлкните левой клавишей мышки.

Изменение размеров графических объектов

Линии, прямоугольники, круги, дуги и фигуры общего вида (paths) при необходимости могут быть масштабированы. Для этого их следует выделить, щёлкнув левой клавишей мышки по краю, и переместить, потянув за прямоугольники «поддержки»:

- Линии имеют две «ручки», соответствующие начальной и конечной точке.
- Прямоугольник имеет восемь «ручек» по числу углов и краёв.
- Круги имеют четыре точки для изменения размера, и все они меняют радиус.
- Дуги имеют две «ручки», что предполагает две конечные и две точки управления Безье.
- Фигуры общего вида имеют по две точки на вершину (точка между двумя сегментами), плюс две точки управления Безье на каждый сегмент кривой.

Другие типы графических объектов не могут менять размер — вместо этого вы должны их удалить и заменить новыми.

Редактирование графических объектов

Все плоские графические объекты могут редактироваться обычным образом после их выделения.

Все плоские графические объекты, исключая графический текст (обсуждение ниже), отображают диалог *Edit Graphics Style*, который позволяет вам задать локальные, фиксированные значения.



См. «Штамп», где описано, как использовать графический текст для отображения информации о проекте.

МАРКЕРЫ

Типы маркеров

Маркеры используются при создании и редактировании устройств, символов, выводов и контактов. Поддерживается следующий фиксированный набор типов:

ТИП	НАЗНАЧЕНИЕ
ORIGIN	Определяет точку прикрепления любого элемента библиотеки. Точка прикрепления — это точка, вокруг которой объект может быть повернут, и согласуется с позицией курсора в момент размещения.
NODE	Определяет позицию точки присоединения провода для контакта или вывода.
BUSNODE	Как и выше, но определяет выводы, как принадлежащие шине. Линия шины (толстая, синяя) будет нарисована от <i>busnode</i> к <i>origin</i> .
LABEL	Определяет позицию и ориентацию этикетки вывода (<i>terminal</i>).
PINNAME	Определяет позицию и ориентацию имени контакта (<i>pin</i>).
PINNUM	Определяет позицию и ориентацию номера контакта.
INCREMENT	Используется для создания моделей симуляции Active Component, этот маркер определяет активный участок для увеличения состояния переменной.
DECREMENT	Как и выше, но для уменьшения состояния переменной.

Размещение маркеров

Маркеры размещаются так же, как и графические символы (каковыми и являются!).

Чтобы разместить маркер:

1. Выберите иконку *2D Graphics Markers Mode* на панели выбора режима.
2. Выберите маркер, который вы хотите разместить, в окне выбора объектов.
3. Используйте иконки *Rotation* и *Mirror*, чтобы придать нужную ориентацию, согласно вашим намерениям по размещению объекта на чертеже. Это справедливо только для *Label*, *Pinname* и *Pinnum* маркеров — ориентация не подходит для других типов.
4. Укажите позицию в окне редактирования, где вы хотите видеть маркер, и щёлкните левой клавишей мышки.

ОСНОВНОЕ О БИБЛИОТЕКАХ

Как задумано, есть две библиотеки символов и около 25 библиотек устройств. Чтобы ознакомиться со списком всех предлагаемых библиотек и частей, прочитайте файл LIBRARY.PDF в директории библиотек вашей установки Proteus. Вам понадобится Adobe Acrobat reader, если его у вас нет.

Строение библиотеки

USERSYM.LIB и USERDVC.LIB установлены в режим чтение/запись; остальные только для чтения. Суть в том, что вы можете добавлять что-то в USERSYM.LIB (новые символы) и USERDVC.LIB (новые устройства). А это означает, что мы можем пополнять остальное через обновления без риска перезаписать объекты со схожими именами в ваших собственных библиотеках.

Вы можете, конечно, создать дополнительные библиотеки, используя *Library Manager*.

Если же вам действительно понадобится что-то поменять в библиотеках, установленных только для чтения, вы можете переустановить их в режим чтения/записи в *File Manager* или проводнике под Windows, или используя *Library Manager* в Proteus.

Ни в коем случае вы не должны удалять что-либо из SYSTEM.LIB.

Команда Pick

Команда *Pick* используется в качестве альтернативы обозревателю библиотек, в первую очередь тогда, когда вы знаете имя устройства или символа, который ищете. Вы можете искать точное соответствие, или также использовать разные шаблоны совпадений, чтобы найти нужное, когда у вас приблизительное представление об имени.

Если вы указываете устройство или символ, который уже загружен, ISIS обновит его из библиотек на диске.

Отметьте также:

- В месте соединения устройств алгоритм замещения ищет соответствия позиции контакта или имён, чтобы соединение сохранилось, даже если вы переместили или перенумеровали контакты устройства.
- Если есть два или более устройств или символов с одинаковыми именами, проходящими через несколько библиотек, команда *Pick* загрузит самые последние. Это обычно полезно, если вы меняете один из ваших элементов в USERDVC.LIB, поскольку ваша версия будет, полагаем, новее нашей.



См. раздел «Отбор, размещение и соединение компонентов», где больше информации о форме библиотечного диалога.

БИБЛИОТЕКИ СИМВОЛОВ

Библиотеки символов используются для поддержки и общих графических символов для непосредственного размещения на чертеже, и также символов выводов, портов модулей и контактов устройств.

Разные типы символов созданы с разными префиксами имён, и, таким образом, существуют для всех целей и назначения, сохраняясь в отдельных «отсеках» библиотеки символов. Набор этих объектов определён в SYSTEM.LIB и предварительно загружается во селекторы объектов, когда вы запускаете ISIS. Таким образом, когда вы выбираете иконку *Terminal*, и вы видите имена DEFAULT, INPUT, OUTPUT и т.д., вы в действительности получаете доступ к символам, названным \$TERDEFAULT, \$TERINPUT, \$TEROUTPUT и т.п. Когда вы помещаете вывод на чертёж, объект *terminal* создаётся, и ему присваивается подходящий символ.

Смысл всего этого ясен из двух соображений:

- Графические символы, выводы, порты модулей и контакты устройств — все хранятся в библиотеках символов (то есть, SYSTEM.LIB и пользовательской USERSYM.LIB). Нет специальных библиотек для выводов, портов или контактов.
- Процедуры для создания разных типов символов очень похожи.

Графические символы

Символы — это группа плоских графических объектов, которые обрабатываются как единственный объект. Например, используя три линии и две дуги вы можете сформировать символ вентиля И.



Чтобы сделать графический символ:

1. Выберите одну из графических иконок, то есть, *Line*, *Box* и т.д.
2. В окне выбора объектов выберите графический стиль подходящий к типу символа, который вы создаёте. Для символа, который формирует базу будущего компонента, это обычно стиль COMPONENT, хотя для некоторых символов, как операционные усилители, где требуются короткие линии, присоединённые к базе, с контактами, лучше использовать стиль PIN.



Если контур символа состоит из линий и дуг, а вы хотите, чтобы символ был закрашен,

используйте фигуру общего вида для создания контура.

3. Выберите и поместите графические объекты, какие требуются для формирования символа.

Любые рисунки, что требуются для придания фиксированного вида, должны редактироваться, соответствующие флажки *Follow Global?* сняты, и атрибуты графического стиля изменены.

4. Если вы хотите определить точку крепления (origin) для символа, выберите иконку *Markers*, щёлкните по маркеру *ORIGIN*, ISIS определит точку крепления в центре символа.
5. Выделите объекты, которые составляют символ, например, прямоугольным выделением.
6. Выполните команду *Make Symbol* из раздела *Library* основного меню, выберите имя и библиотеку для нового символа.
7. Щёлкните по **ОК**, чтобы завершить операцию.

Символ может состоять только из плоских графических объектов — вы не можете выделять всю секцию схемы, включающую компоненты, соединения и т.п., чтобы превратить это в символ. Чтобы манипулировать частями схемы подобным образом, вы должны использовать команды *Import* и *Export section* раздела *File* основного меню.

Определённые пользователем выводы (Terminals)

ISIS допускает определение пользовательских символов для использования в качестве логических или физических выводов. Все осуществляется таким же образом, как и с обычными символами, за исключением того, что вы должны разместить маркер *Node*, чтобы задать позицию точки соединения вывода, и маркер *Label* для задания положения и ориентации этикетки цепи.

Чтобы создать определённый пользователем вывод:

1. Выберите подходящую иконку графического элемента — обычно это иконка *Line* — на панели *выбора режима*.
2. В окне *выбора объектов* выделите подходящий графический стиль. Как правило, это будет всегда стиль *TERMINAL*, хотя и стиль *BUS WIRE* может понадобиться для небольших элементов символа.
3. Выберите и разместите графические объекты нужным образом для формирования основания вывода.

Любая графика, требующая фиксированного вида, должна редактироваться, соответствующие флажки *Follow Global?* сниматься, а атрибуты графического стиля меняться.

4. Выберите иконку *Markers*. Поместите *Node* или *Busnode* маркер там, где к выводу будет

присоединяться провод или шина, а маркер *Label* там, где вы хотите видеть этикетку цепи. Вам нужно добавить маркер *Origin*, чтобы определить, где будет точка установки символа.

5. Выделите объекты, составляющие вывод, например, прямоугольным выделением.
6. Используйте команду *Make Symbol* из раздела *Library* основного меню, задайте тип для *Terminal* и затем выберите имя и библиотеку для нового вывода.
7. Щёлкните по **ОК**, чтобы завершить операцию.

Заметьте, что электрический тип определённых пользователем выводов всегда по умолчанию пассивный (*passive*). Если вам нужно задать другой электрический тип, вы должны назначить его после размещения с помощью *Property Assignment Tool*.

Определённые пользователем порты модуля

Порты модуля — это выводы, используемые для соединения проводов с подсхемами, и допускается для этой цели использовать определённые пользователем символы. Все выполняется так же, как и с обычными символами, за исключением того, что вы должны разместить маркер *Node*, чтобы задать позицию точки соединения вывода, и маркер *Label* для задания положения и ориентации этикетки цепи.

Чтобы создать определённый пользователем порт модуля:

1. Выберите подходящую иконку графического элемента — обычно это иконка *Line* — на панели выбора режима.
2. В окне выбора объектов выделите подходящий графический стиль. Как правило, это будет всегда стиль *PORT*, хотя и стиль *BUS WIRE* может понадобиться для небольших элементов символа.
3. Выберите и разместите графические объекты нужные для символа контакта (*pin*).

Любая графика, требующая фиксированного вида, должна редактироваться, соответствующие флажки *Follow Global?* сниматься, а атрибуты графического стиля меняться.

В рисунке порта вы должны ориентировать его так, чтобы удобно было размещать его с левой стороны подсхемы — ISIS отразит его по X, когда он будет размещаться справа.

4. Выберите иконку *Markers*. Поместите *Node* или *Busnode* маркер там, где к порту будет присоединяться провод или шина, а маркер *Label* там, где вы хотите видеть этикетку цепи. Вам нужно добавить маркер *Origin*, чтобы определить, где будет край подсхемы.
5. Выделите объекты, составляющие вывод, например, прямоугольным выделением.
6. Используйте команду *Make Symbol* из раздела *Library* основного меню, задайте тип для *Module Port* и затем выберите имя и библиотеку для нового вывода.
7. Щёлкните по **ОК**, чтобы завершить операцию.

Определённые пользователем контакты (pins) устройства

Контакты устройства, фактически, рисуются как символы и, следовательно, вы можете определить собственные символы для них. Большой набор символов контактов есть в SYSTEM.LIB.

Тем не менее, возможны ситуации, когда предпочтительнее собственные.

Чтобы сделать определённый пользователем контакт устройства:

1. Выберите подходящую иконку графического элемента — обычно это иконка *Line* — на панели выбора режима.
2. В окне выбора объектов выделите подходящий графический стиль. Как правило, это будет всегда стиль PIN, хотя и стиль BUS WIRE может понадобиться для небольших элементов символа.
3. Выберите и разместите графические объекты нужные для символа контакта (pin).
Любая графика, требующая фиксированного вида, должна редактироваться, соответствующие флажки *Follow Global?* сниматься, а атрибуты графического стиля меняться.
4. Выберите иконку *Markers*. Поместите *Node* или *Busnode* маркер там, где к порту будет присоединяться провод или шина, а маркер *Pinname* там, где вы хотите видеть этикетку контакта и маркер *Pinum*, где должен быть номер вывода. Вам нужно добавить маркер *Origin*, чтобы определить, где будет точка установки символа.
5. Выделите объекты, составляющие вывод, например, прямоугольным выделением.
6. Используйте команду *Make Symbol* из раздела *Library* основного меню, задайте тип для *Device Pin* и затем выберите имя и библиотеку для нового вывода.
7. Щёлкните по **ОК**, чтобы завершить операцию.

Редактирование существующего символа

Любой тип символа можно редактировать, поместив его образец и затем используя команду *Decompose* из раздела *Library* основного меню.

Чтобы редактировать символ:

1. Поместите образец символа. Это будет *Graphics Symbol*, *Terminal*, *Module Port* или *Device Pin*, как удобнее.
2. Выделите объект щелчком по нему и щёлкните правой клавишей мышки.
3. Выберите команду *Decompose* из выпадающего меню. Этим вы разделите символ на графику и маркеры.
4. Пополните, удалите или отредактируйте графику и маркеры, как требуется.

5. Реконструируйте символ согласно процедуре из предыдущих разделов, соответственно типу.

Определение иерархических символов

ISIS довольно успешно позволяет символам содержать другие символы и/или другие графические объекты. Это позволяет вам создать, например, вентиль NAND из ранее определённого вентиля AND, добавив кружок. Заметьте, что символы, определённые из других символов, не «связаны» с ними никак — если символ нижнего уровня изменён или даже удалён, это не скажется на вновь созданном. Это также означает, что символ может быть определён, как модификация самого себя без особых трудностей, хотя это разрушит старую версию.

БИБЛИОТЕКИ УСТРОЙСТВ

Устройство (device), в терминологии ISIS, это тип реального компонента, такого как транзистор NPN или микропроцессор PIC. Из этого следует, что компонент, помещённый на чертёж, есть образец устройства. По существу, есть три типа устройств:

- Одно-элементное устройство. Это компоненты, для которых есть однозначное соответствие между символом схемы и корпусом PCB. Каждый вывод имеет единственное имя и единственный номер вывода.
- Гомогенные (однородные) много-элементные устройства. Это компоненты, для которых есть несколько идентичных элементов в одном корпусе. Типичный пример — это четыре вентиля NAND 7400 серии или два ОУ TL072. Одинаковые выводы имеют разные номера ножек для каждого элемента, исключая выводы питания, остающиеся общими.
- Гетерогенные (разнородные) много-элементные устройства. Есть компоненты, где есть несколько разных элементов в одном PCB корпусе, но при этом вы хотите отобразить каждый элемент отдельно на схеме. Самым общим примером может служить реле, когда вам нужно отдельно изобразить обмотку и одну или несколько групп контактов в разных местах схемы.

ISIS также предлагает поддержку для выводов шины. Устройства, такие как микропроцессоры и их периферия, могут быть нарисованы в очень компактном виде, поскольку их шины данных и адреса могут быть представлены единственным выводом. Это также облегчает их соединение.

Тогда как одно-элементное устройство будет иметь только один физический номер вывода, ассоциированный с каждым схемным выводом, много-элементные компоненты и компоненты с выводами шины, все они имеют множество номеров выводов для каждого из контактов устройства. Этот аспект процесса создания устройства поддерживается Visual Packaging Tool. Дополнительно этот инструмент позволит вам создавать альтернативные упаковки (каждая из которых с собственными номерами выводов) для одинаковых элементов схемы. Типичным примером этого может быть микропроцессорное устройство, как PIC16F877, которое может иметь корпус и DIL40, и PLCC44.

Создание элемента устройства

Большинство устройств, которые вы встретите, будут содержать только один элемент. Это, так сказать, размещением одного компонента вы получаете доступ ко всем контактам физического объекта. Полный контраст с много-элементными компонентами, подобными 7400, для которых вам нужно разместить четыре вентиля, чтобы получить доступ ко всем контактам. Другими словами, первый этап в создании нового элемента библиотеки — это размещение графики и контактов для элемента устройства или элементов.

Чтобы создать элемент устройства:

1. Разместите графические объекты, чтобы определить базу устройства в подходящем графическом стиле(ях).
2. Разместите контакты объектов для представления выводов.
3. Пометьте выводы, присвоив им имена и типы, используя любую стандартную технику редактирования (см. «Редактирование объекта») или *Property Assignment Tool*.
4. Выделите все объекты, образующие элемент. Затем вызовите команду *Make Device* и присвойте свойства по умолчанию.

Эти этапы подходят для дальнейшего обсуждения:

Определение базы устройства

База устройства, в сущности, полный рисунок устройства, исключая его ссылочный указатель и контакты. Очень часто это только прямоугольник, и в этом случае вам понадобится просто выбрать и разместить подходящего размера прямоугольник в графическом стиле COMPONENT. Для компонентов с более сложной графикой, как транзисторы, операционные усилители и т.д., вы можете использовать любые графические объекты, которые предлагает ISIS, в любом подходящем графическом стиле, и, возможно, отредактировать объекты и придать локальные, фиксированные значения некоторым или всем атрибутам графического стиля.

Очень важно тщательно продумать, какой графический стиль вы выберете для графики нового компонента. В основном графические элементы будут использованы со стилем COMPONENT, и графические объекты не будут нуждаться в редактировании, поскольку они полностью предопределены, следуя родительскому стилю COMPONENT. Но изредка вам понадобится нарисовать такую часть графики нового устройства, которая должна быть фиксирована (fixed). Например, вам может понадобиться закрасить транзистор, чтобы он всегда был чёрным. В этом случае разумно отредактировать графический объект после размещения, снять некоторые или все флажки атрибутов графического стиля и задать локальные значения, хотя, где можно, вы должны стремиться отставить атрибуты родительского стиля COMPONENT. Вашей следующей заботой должна стать подгонка размещения графического объекта на схеме. Например, если вы используете операционный усилитель, вам захочется использовать короткие линии между косыми линиями базы фигуры и выводом питания; такие выводы лучше рисовать линиями в стиле PIN, в этом случае они выглядят, как часть контакта.

Если вы хотите определить точку установки (origin) для устройства, вы должны также поместить маркер *Origin* в подходящем месте. Если вы не зададите его, ISIS определит его на

конце верхнего левого вывода.



См. раздел «Графика» о том, как размещать и редактировать графические объекты.

Размещение выводов

Когда вы создаёте устройство, вы используете специальные объекты, контакты устройства, доступные с помощью иконки *Device Pin*, и размещаете их по очереди. Несколько типов выводов (pin) устройств предзагружаются в окно выбора, когда вы в первый раз используете иконку *Device Pin*, а в дальнейшем типы выводов можно указать, если требуется, в библиотеках символов. Вы можете также определить собственные типы выводов.

Когда вы размещаете объект контакт, вам нужно быть уверенным, что он сориентирован правильно. Синее перекрестие, которое появляется с одной стороны каждого вывода, означает точку соединения для вывода; остальное будет обычно в контакте с некоторыми частями основания устройства.



Больше информации об объектах «контакты» можно найти в соответствующем разделе.

Обозначение выводов

Это третья, возможно, самая мудрёная фаза процесса создания устройства, и может стать причиной появления неясных проблем позже, при разработке, если она выполнена неправильно. Вас предупредили!

Каждый вывод может нести имя, номер контакта и электрический тип. Последний используется при проверке правильности соединений (electrical rules checking, ERC) и симулятором ProSPICE. Электрический тип вывода должен быть корректно задан для моделей цифрового симулятора, в частности.

У вас есть два основных подхода к обозначению выводов:

- Последовательно редактировать каждый вывод (укажите маркером и нажмите **CTRL+E**, что легче) и продолжить редактирование, используя диалоговую форму свойств.
- Использовать *Property Assignment Tool* для доступа к свойствам PINNAME, PINNUM и TYPE выводов.

В большинстве случаев вы найдёте, что удобно использовать смешанные техники.

При присвоении имён и номеров выводам имейте в виду следующее:

- Выводы должны всегда иметь имена. Если вы вводите номер при отсутствии имени, вывод автоматически получит имя то же, что и его номер.
- Если вы даёте двум или более выводам одинаковые имена, они будут считаться электрически соединёнными в netlist и при разводке печатной платы.
- При размещении выводов с двойным назначением в их именах используйте знак доллара (\$), чтобы отметить первое и второе значение. Например, RD/\$WR\$ будет отображено, как RD/WR.

LABCENTER ELECTRONICS

- В общем, легче присвоить номера выводам, используя Visual Packaging Tool, и это единственный путь присвоить номера выводов для много-элементных компонентов или компонентов с выводами шины. Однако для простых единичных элементов устройств вы можете ввести номера выводов на этом этапе, если хотите.

При присвоении типов выводов следующая таблица может оказаться полезна:

Pin Type (тип вывода)	TYPE ID (идентификатор)	Пример использования
Passive	PS	Пассивный вывод устройства
Input	IP	Аналоговый или цифровой вход
Output	OP	Аналоговый или цифровой выход
Bidir	IO	Микропроцессорные или оперативной памяти выводы шины
Tri-state	TS	Выходы ПЗУ
Pull Down	PD	Выходы с открытым коллектором или стоком
Pull Up	PU	Выходы с открытым эмиттером или истоком
Power	PP	Выводы питания и земли

Если вам не ясно, как фактически применить требуемые операции редактирования, следующие разделы руководства вам помогут:



См. раздел «Объекты выводы» о диалоговой форме редактирования контактов.



См. раздел «Property Assignment Tool» с примерами использования этого инструмента.

Вызов команды *Make Device*

Заключительный этап создание единственного элемента устройства — это выделить все объекты (графические и контакты), составляющие устройство, и затем вызвать команду *Make Device* в разделе *Library* основного меню. Использование этой команды обсуждается в деталях в следующих разделах.

Команда *Make Device*

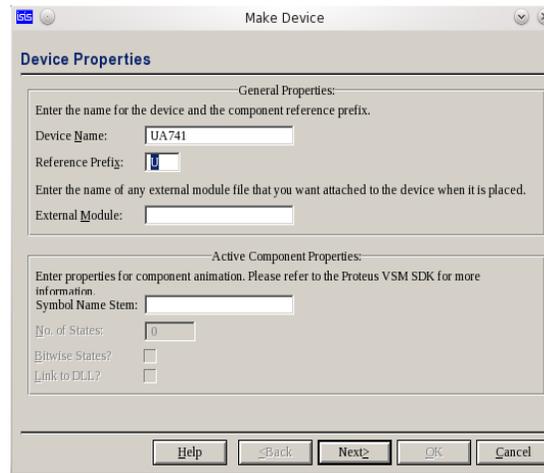
Команда *Make Device* — это многоэтапная диалоговая форма, иногда называемая Wizard (помощник) в других приложениях. Есть четыре страницы:

- Свойства устройства (Device properties) — есть такие вещи, как имя устройства, префикс для использования новых компонентов и также свойства, ассоциированные с анимацией компонентов, которые используются в Proteus VSM.
- Упаковка (Packaging, корпус) — эта страница отображает упаковку, которая была предопределена (для существующего компонента) и предоставляет доступ к *Visual Packaging Tool*.
- Свойства компонента (Component Properties) — эта страница предлагает средство создания и редактирования определения свойств и значений свойств.
- Выбор библиотеки (Library selection) — последний экран, позволяющий вам выбрать библиотеку, в которой устройство будет храниться.

Детальная контекстно чувствительная подсказка предлагается для всех полей этих страниц. Если вы не уверены в назначении отдельных полей, щёлкните по «?» в верхней правой части диалога и затем щёлкните по полю, чтобы увидеть объяснение его назначения. Сделав это, мы ограничили разъяснения здесь до общей информации о каждой странице.

Страница свойств устройства

Эта страница имеет две основные секции: *General Properties* и *Active Component Properties*.

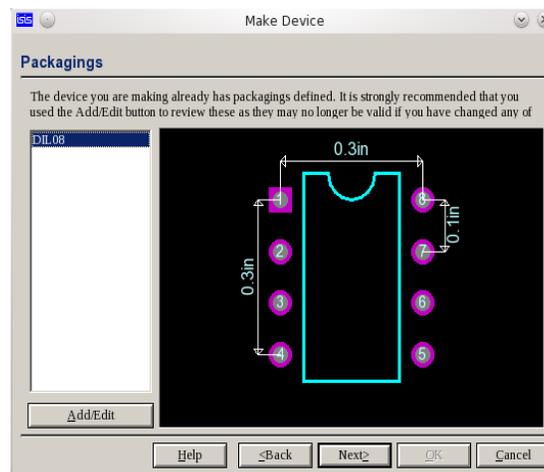


Секция *General Properties* определяет имя устройства и ссылочный префикс. Это буква или буквы, которые появляются перед идентификатором элемента для вновь размещаемых компонентов. Заметьте, что, если вы оставляете это поле пустым, вновь размещаемые компоненты будут не аннотированы и что значение их части и текст свойств будут также скрыты. Это полезно для рисунков блочных диаграмм или компонентов, как виртуальный осциллограф, которые не являются реальными частями проекта.

Секция *Active Component Properties* используется для создания анимированных компонентов, используемых с Proteus VSM. Посмотрите документацию к Proteus VSM SDK, где больше информации.

Страница упаковки (Packagings)

Эта страница отображает набор корпусов, которые были определены для устройства; для новых устройств список будет пуст.

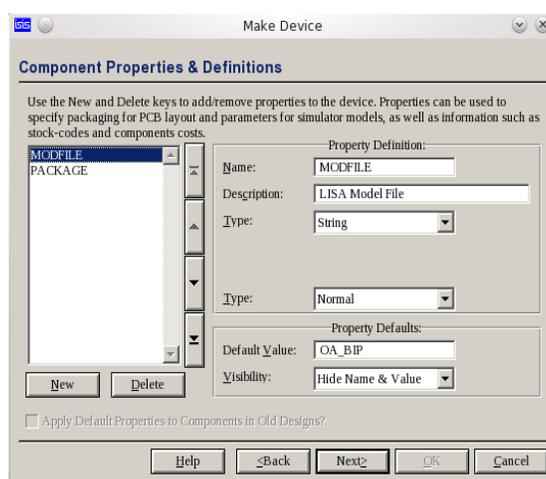


Нажатие кнопки **Add/Edit** вызовет *Visual Packaging Tool*, который описан подробнее в соответствующем разделе.

Заметьте, что требуются иные процедуры для упаковки гетерогенных много-элементных устройств. В этом случае инструмент упаковки должен вызываться для полного набора элементов, расположенных на схеме. См. соответствующее описание.

Страница свойств и определений (Component Properties&Definitions)

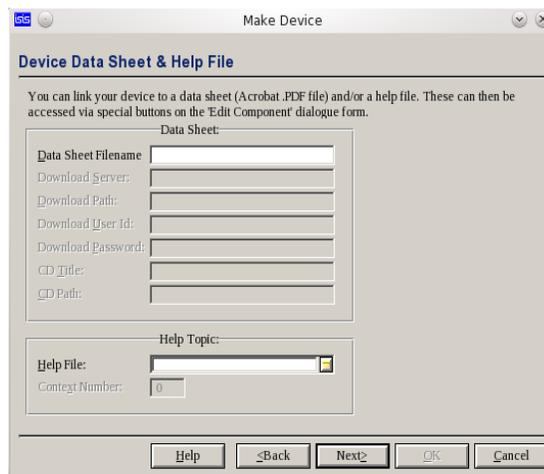
Эта страница используется для определения свойств и определения значений для свойств компонента. Селектор слева показывает свойства, которые были определены, тогда как секции *Property Definition* и *Property Defaults* определяют тип, видимость и предопределённые значения свойств.



См. «Определение свойств», где больше информации о свойствах компонентов и их определении.

Страница справочного листа и подсказки (Data Sheet and Help)

Эта страница позволяет вам ассоциировать справочный лист (PDF файл) и/или раздел подсказки с устройством. Если справочный лист определён, то в диалоге *Edit Component* появится кнопка **Data**, а если определена подсказка, то и кнопка **Help**. Вы увидите эти кнопки для многих компонентов в предложенных библиотеках.



Справочные листы (Data sheets) могут располагаться в трёх местах:

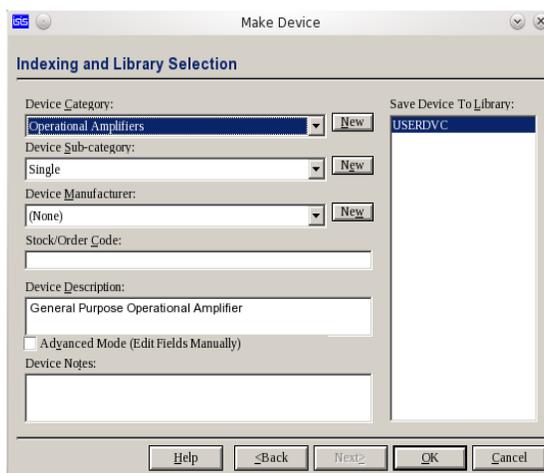
- В директории Data вашей инсталляции Proteus. Вы можете изменить место, где Proteus будет искать справочный лист, щёлкнув по кнопке **Path Settings** в диалоге *Data Sheet Not Found*, который появляется, когда Proteus не может найти справочный листок.
- На FTP сервере — либо в Интернете, либо на сервере вашей компании. Вам может понадобиться ввести идентификатор и пароль пользователя для некоторых серверов.
- На CD или CD-г.

Основные соображения в том, что есть центральное хранилище справочных листков, а Proteus скопирует их для индивидуальных пользователей на основе их потребностей.

Ссылки на разделы подсказок наиболее полезны, если вы создаёте комплексные модели для симулятора ProSPICE, и нужно создать документацию, связанную с моделью. Это, скорее всего, существенно для опытных пользователей и разработчиков моделей.

Страница выбора библиотеки

Последняя страница позволяет вам выбрать, в какой библиотеке новое устройство будет храниться. Только библиотеки для чтения/записи (в противоположность тем, что только для чтения) могут быть отображены.

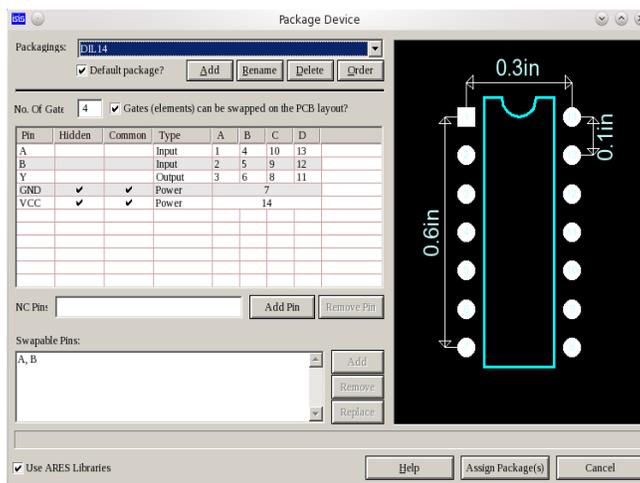


Как только вы щёлкните по кнопке **OK**, устройство будет сохранено в выбранной библиотеке. Также, если оно есть в проекте, вы получите приглашение обновить все компоненты, которые используют его, с новым определением.

Visual Packaging Tool

Визуальный инструмент упаковки предлагает графическое окружение, в котором присваиваются одно или более PCB footprints (контактные площадки под компонент для печатной платы) для элемента схемы. Для каждого корпуса PCB создаётся таблица картирования номеров выводов и имён выводов, так что разные упаковки могут иметь разные номера выводов для одинаковых выводов в схеме.

Инструмент упаковки также облегчает ввод разных номеров выводов для каждого элемента в много-элементных компонентах и каждого вывода в шине.



Инструмент упаковки может вызываться двумя способами:

LABCENTER ELECTRONICS

- Щелчком по кнопке **Add/Edit** страницы Packaging в диалоге *Make Device*.
- Размещением элемента или элементов устройства, для которого нужен корпус, в схеме, выделением его и выбором команды *Packaging Tool* в разделе *Library* основного меню.

Второй способ — единственный путь для создания упаковки гетерогенных много-элементных компонентов.

Как и с командой *Make Device* детальная контекстно чувствительная подсказка доступна для всех полей. Чтобы увидеть подсказку в отдельном поле, щёлкните по «?» сверху справа на форме, а затем щёлкните собственно по полю.

Packaging Selector (селектор упаковки)

Секция *Packages* инструмента упаковки показывает список-селектор для корпусов, которые могут быть пока определены, и кнопки для добавления, переименования, удаления и упорядочивания их. Один корпус может быть выбран в качестве предустановленного, и это тот, который будет использоваться для вновь размещаемых компонентов.

При условии, что *ARES* установлена, кнопка **Add** будет вызвать копию обозревателя библиотеки упаковок, позволяя вам выбрать контактные площадки (footprint) для нового корпуса.

Сетка выводов

Основная работа инструмента упаковки — это вывести, используя сетку выводов. Здесь вы можете ввести номера выводов для каждого контакта и каждого элемента, и также выбрать, какие выводы будут общими для элементов много-элементного устройства.

Электрический тип отображается только для информации, и здесь его нельзя менять — вы должны определить его, когда размещаете индивидуальные выводы для элемента устройства.

Количество элементов

Это поле определяет количество колонок с номерами выводов, которые появляются в сетке выводов. Для много-элементных гетерогенных компонентов, оно должно представлять общее количество элементов.

Обозреватель упаковки

Обозреватель упаковки отображает контактные площадки, выбранные для текущей упаковки. Поскольку выводы присвоены сетке выводов, все подсвечено так, что вы можете видеть, какие выводы остаются свободными. Также, если вы укажете вывод мышкой, *ISIS* отобразит ассоциированное с ним имя (если оно есть) и номер вывода.

Если текстовый курсор на сетке выводов, тогда щелчок по выводу в обозревателе введёт его номер в сетку выводов и вывод будет подсвечен, чтобы показать, что он теперь определён.

Скрытые выводы

Скрытые выводы в основном используются для задания питания компонентов так, чтобы не перегружать схему лишними линиями. Скрытый вывод может быть определён одним из двух способов:

- Сбросом флажка *Draw Body* в диалоге *Edit Pin*. Это должно быть сделано, когда создаётся устройство. Этот вывод появится в Pin Grid (сетка выводов), когда вызывается инструмент упаковки, и он будет представлен во всех корпусах.
- Щелчком по кнопке **Add Hidden Pin**. Этим создаётся скрытый вывод, который будет характерен для данной упаковки и неявно общим для всех её элементов.

Общие выводы

Общий вывод (common pin) — это один из тех, что имеют тот же номер вывода для всех элементов много-элементного устройства. Обычно это относится к выводам питания и выводам разрешения/стробирования, которые являются общими для всех буферов в много-элементном драйвере.

Если общий вывод не скрытый, тогда любой провод, подходящий к нему в одном элементе, будет присоединён к другим элементам.

Общий вывод также считается связанным со всеми элементами в гетерогенном много-элементном компоненте, даже если он не представлен на каких-то из них. Это имеет последствия для правил замены вентиляей (gate swap) в ARES.

Замена вентиляей

Установка этого флажка сообщит ARES, что вентиляльные элементы могут быть заменены.



См. соответствующий раздел о замене выводов и вентиляей.

Заменяемые выводы

Секция заменяемых выводов позволяет вам определить группы выводов, которые электрически взаимозаменяемы. Например, два входа A и B счетверённого вентиля NAND серии 7400 электрически идентичны, вследствие чего могут заменяться на плате для удобства разводки печати.

Вы можете добавлять группы взаимозаменяемости в список, подсвечивая выводы в сетке выводов (удержите клавишу **CTRL** и щёлкните по именам выводов), и затем щёлкнув по кнопке **Add**.



См. соответствующий раздел о замене выводов и вентиляей.

Не присоединенные выводы (NC Pins)

При создании корпуса большого устройства может оказаться полезным «убедиться», что все выводы были учтены, проверив, что все выводы в обозревателе корпуса подсвечены. Вместе с тем, некоторые выводы могут задаваться как свободные (не присоединённые), это полезно, чтобы записать их при подсчёте. Осуществляется это вписыванием их номеров в поле NC Pins. Применяйте запятые для разделения номеров выводов.

Создание одно-элементных устройств

Большая часть устройств, которые вы будете создавать, будут иметь однозначное соответствие между символом схемы и корпусом на плате. В этом случае процедура почти совпадает с той, что при создании элементов устройств, описанных выше.

Чтобы создать устройство с единственным элементом:

1. Разместите графические объекты, чтобы определить основание устройства в подходящем графическом стиле(ях).
2. Разместите контакты устройства, чтобы представить его выводы.
3. Пометьте контакты, задав им имена и типы, используя для этого любую стандартную технику редактирования (см. «Редактирование объектов») или *Property Assignment Tool*.
4. Выделите все объекты, составляющие устройство. Затем используйте команду *Make Device*.
5. Введите имя и префикс устройства на странице Device Properties, затем щёлкните по кнопке **Next**, чтобы перейти на страницу Packaging.
6. При условии, что устройство имеет контактные площадки (footprint) для разводки, щёлкните по кнопке **Add/Edit** и используйте *Visual Packaging Tool*, чтобы задать упаковку и нумерацию выводов. Когда все сделано, щёлкните по кнопке **Assign Packages**.
7. Щёлкните по кнопке **Next**, чтобы перейти к странице Component Properties и ввести любые определения свойств и значения для других свойств компонента.
8. Щёлкните по кнопке **Next**, чтобы перейти на страницу Data Sheet и задать место хранения справочного листка или раздела подсказки, связанных с устройством.
9. Щёлкните по кнопке **Next**, чтобы перейти на страницу Library Selection и выбрать библиотеку, в которой хотите хранить ваше устройство.
10. В завершение щёлкните по кнопке **OK**, чтобы завершить процесс.

Если вы предпочитаете, вы можете также назначить номера выводов для единственного элемента устройства на шаге [3]. В этом случае номера выводов появятся автоматически в *Visual Packaging Tool*.

Создание много-элементного гомогенного устройства

Много-элементное гомогенное устройство похоже на 7400, в котором физические части состоят из нескольких идентичных элементов, и которые вы можете размещать как отдельные компоненты на схеме. Для поддержки таких устройств ISIS должен позволять использовать разные наборы номеров выводов, которые присваиваются одинаковым элементам. Для 7400, где, например, может быть четыре набора для номеров выводов — по одному для каждого вентиля — если вы маркировали вентиль 7400, как U1:C, тогда ISIS должен использовать третий набор выводов: 8, 9 и 10. Это все поддерживается *Visual Packaging Tool*.

Чтобы сделать много-элементное гомогенное устройство:

1. Создайте основание устройства и разместите его выводы, как обсуждалось выше.
2. Пометьте выводы, добавив имена и тип, используя любую стандартную технику редактирования (см. «Редактирование объектов») или *Property Assignment Tool*.
3. Выделите все объекты, составляющие устройство. Затем используйте команду *Make Device*.
4. Введите имя и префикс устройства на странице *Device Properties*, затем щёлкните по кнопке **Next**, чтобы перейти на страницу *Packaging*.
5. Щёлкните по кнопке **Add/Edit** на странице *Packaging*, чтобы вызвать *Visual Packaging Tool*.
6. Задайте номер элемента в устройстве и затем заполните поля номеров выводов для каждого элемента поочерёдно. Установите флажки в колонке *common*, чтобы обозначить выводы (как выводы питания), которые имеют одинаковые номера для всех элементов.
7. Продолжайте все, как для единственного элемента, начиная с шага [7].

Создание много-элементного гетерогенного устройства

Много-элементные гетерогенные устройства определены как элементы, состоящие из нескольких разных частей, каждая из которых может располагаться, как отдельный компонент, на чертеже. Наиболее общий пример, возможно, это реле, где вы имеете обмотку и одну или более групп контактов. Требование таково — размещение обмотки в одной точке чертежа, а набор(ы) контактов где-то ещё.

Как и с много-элементным гомогенным устройством, используется *Visual Packaging Tool*, чтобы распределить номера выводов для каждого элемента.

Чтобы создать много-элементное гетерогенное устройство:

1. Поместите графические объекты и выводы устройства, чтобы определить вид элементов. Удостоверьтесь, что элементы полностью разделены в *окне редактирования*, так что вы можете выделять объекты, так что включается только он сам.

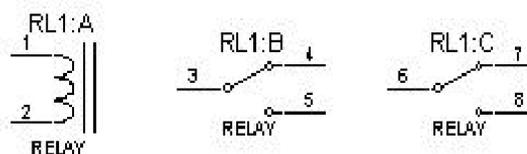


Процесс создания элемента устройства описан выше.

2. Пометьте выводы, задав только имена и тип.
3. Выделите все объекты, которые составляют первый элемент. Затем вызовите команду *Make Device* из раздела *Tools* основного меню. Введите Device Name в форме:
NAME:A
где NAME — это то, как вы хотите назвать часть в целом. Суффикс A говорит ISIS, что это первый элемент гетерогенного устройства.
4. Щёлкните по кнопке **Next** дважды, чтобы перейти к странице Component Properties и ввести любые определения свойств и определённые значения для других свойств компонента. Заметьте, что вы на этом этапе вы не вызываете для гетерогенных частей инструмент упаковки.
5. Щёлкните по кнопке **Next**, чтобы перейти к странице Data Sheet и задать местоположение любых справочных листков или разделов подсказки, которые вы хотите связать с устройством.
6. Щёлкните по кнопке **Next**, чтобы перейти к странице Library Selection и выбрать библиотеку, в которой вы намерены хранить элемент устройства и щёлкните по **OK**, чтобы сейчас сохранить его.
7. Повторите процедуру с шага [3] для других элементов, давая им имена, как
NAME:B, NAME:C
и т.д.
8. Выберите иконку *Component* и поместите один образец каждого элемента устройства на свободное место схемы.
9. Выделите размещённые элементы и затем вызовите *Packaging Tool* из *Library* основного меню. Вы должны увидеть, что есть колонка номеров выводов для каждого элемента, но что выводы, которые недоступны для отдельных элементов имеют «---» в колонке номера.
10. Создайте корпус и введите нумерацию выводов обычным образом.
11. Щёлкните по кнопке *Assign Packages*, чтобы сохранить упаковку в библиотеке. Заметьте, что элемент библиотеки, который вы создали для каждого элемента устройства будет обновлён.

Пример

Поскольку это довольно сложный процесс, мы поясним его на простом примере. Рассмотрите обмотку реле и контакты, как показано ниже.



После создания элементов устройства RELAY:A, RELAY:B и RELAY:C вы вызываете Visual Packaging Tool и задаёте сетку выводов следующим образом:

Pin	Hidden	Common	Type	A	B	C
C1			Passive	1	---	---
C2			Passive	2	---	---
COM			Passive	---	3	6
NC			Passive	---	4	7
NO			Passive	---	5	8

Создание устройства с шинными выводами

Система библиотек устройств ISIS имеет возможность поддерживать устройства, в которых единственный вывод на рисунке компонента представляет несколько выводов физического элемента. Такой вывод называется шиной выводов и предназначен для эффективного представления микропроцессоров и их поддерживаемых чипов. Как и для много-элементных устройств, процесс введения номеров выводов для каждого бита шины выполняется в *Visual Packaging Tool*.

Чтобы создать устройство с шинными выводами:

1. Создайте основание устройства и разместите его выводы так же, как и для устройства с единственным элементом. Вы должны использовать *Bus pin* или определённые пользователем выводы с *Busnode* маркером для выводов шины.
2. Пометьте выводы, задав только имена и тип. Выводам шины должна присваиваться полная спецификация шины, как D[0..7]. Однако вы можете сделать диапазон значений шины скрытым — то есть, часть [0..7] — в схеме, сняв флажок *Draw Bus Range*.
3. Выделите все объекты, составляющие устройство и вызовите команду *Make Device*.
4. Введите имя и ссылочный префикс для устройства на странице *Device Properties*, затем щёлкните по кнопке **Next**, чтобы перейти на страницу *Packaging*.
5. При условии, что устройство имеет контактные площадки (*footprint*) для разводки, щёлкните по кнопке **Add/Edit** и используйте *Visual Packaging Tool*, чтобы задать упаковку и нумерацию выводов. Вы увидите, что каждый бит шины имеет свою строку в сетке выводов. Благодаря этому, процесс присвоения номеров выводам полностью то же, что и для единственного элемента.

Когда все сделано, щёлкните по кнопке **Assign Packages**.

6. Продолжайте с шага [7] для создания устройства с единственным элементом.

Определение свойств и свойства по умолчанию

Есть множество приложений, в которых полезно устраивать элементы библиотеки так, чтобы вносить определённые пользователем свойства, которые будут автоматически присваиваться каждому компоненту во время его размещения. Несколько общих примеров включают:

- PACKAGE свойства для разводки печатной платы. Большинство библиотечных элементов в представленных библиотеках вносят эти свойства предопределёнными для сквозных отверстий корпуса.
- PRIMITIVE, MODEL и MODFILE свойства для симулятора VSM. Вновь, все наши элементы библиотеки для каждого, поддерживаемого симулятором, вносят эти предустановленные свойства.
- Складские коды и коды поставщика. Если вы добавили определение свойств для них в ваши библиотеки, они могут быть включены в спецификацию Bill of Materials.

Чтобы добавить определённые свойства устройству:

1. Следуйте шагам раздела «Создание элемента устройства».
2. Когда вы дойдёте до формы диалога *Make Device*, используйте страницу Component Properties, чтобы присвоить определение свойств и определённых значений свойствам.
3. Сохраните устройство в библиотеке обычным образом.

Для добавления или изменения свойств существующего устройства:

1. Выберите, поместите и выделите образец устройства. Нет нужды разбирать (decompose) его.
2. Вызовите команду *Make Device* и щёлкните кнопку **Next**, чтобы перейти на страницу Component Properties. Все существующие определения свойств будут отображены.
3. Отредактируйте свойства, как требуется.
4. Щёлкните **Next** дважды и затем сохраните устройство в библиотеке, щёлкнув по **ОК**.

Определённые свойства могут также присваиваться элементам библиотеки массировано, если применить скрипт ADI в Library Manager.

Распределение выводов питания (Power Pins)

Поддержка выводов питания тяготеет к какому-то сбивающему с толку субъекту с разным вводом в схему упаковок, выполняющих разные вещи. Дальнейшие трудности возникают при необходимости аккуратно определять, что случится с шиной питания, пока чертёж не «перегрузится» тем, что в значительной мере есть тривиальная информация.

ISIS предоставляет вам разнообразные возможности для поддержки выводов питания:

- Сделайте выводы питания видимыми и физически соедините их с подходящими точками электрической цепи. В этом есть преимущество — вы можете видеть, что и где соединено, но есть тенденция к появлению «мешанины», когда в схеме много интегральных схем.
- Спрячьте выводы питания и позвольте ISIS определить их соединение через именованные сети. Скрытый вывод VCC будет в этом случае соединён с VCC сетью, а скрытый вывод GND (земля) с сетью GND.
- Спрячьте выводы питания и задайте явно, с какой сетью они соединены, используя именованные пользовательские свойства. Например, пользовательское свойство $VCC = +5V$ применится к объекту со скрытым VCC выводом, и, следовательно, этот вывод будет соединён с +5V net.

Вы можете, конечно, использовать смешанные техники, как удобнее, для разных частей чертежа. В общем, мы находим, что использование скрытых выводов питания почти необходимо для цифровых разработок, тогда как в аналоговой работе, порой, проще соединять выводы питания с подходящей шиной. Это положение дел отображено в конструкции предоставляемых элементов библиотек.

Чтобы создать скрытый вывод на простом единственном элементе:

1. Поместите объект вывод для контакта обычным путём.
2. Вызовите диалог *Edit Pin*, щёлкнув по выводу правой клавишей и выбрав команду из выпадающего меню.
3. Снимите флажок *Draw body*. Это скроет вывод, имя и номер безотносительно состояния флажков *Draw name* и *Draw number*.
4. Если вывод — вывод питания, установите его тип как Power.

Наличие объекта контакт (pin) затем будет маркировано только темно-синим крестиком на конце узла, а вывод не будет рисоваться для всех или некоторых устройств, в которые он включен. При создании netlist, он будет автоматически соединён с сетью с тем же именем, что и имя вывода, пока не будет явно переписан соответствующим образом названным пользовательским свойством.

Чтобы создать скрытый вывод питания в корпусе:

1. Создайте элемент(ы) устройства обычным образом, но не помещайте объект вывод для скрытого контакта.
2. При создании упаковки используйте кнопку **Add Pin** для создания дополнительной строки в сетке выводов для скрытого вывода.
3. Впечатайте имя для скрытого вывода, как VCC, а затем присвойте выводу номер.

LABCENTER ELECTRONICS

Как и с другими типами скрытых выводов, при создании netlist ISIS соединит все скрытые выводы, заданные в упаковке, с сетями с тем же именем, что дано и выводу, пока вы не измените это с помощью соответственно названного пользовательского свойства.

Чтобы изменить выводы скрытой сети:

1. Выделите все элементы компонента, у которого есть скрытые выводы. Проще всего этого добиться, используя команду *Search and Tag* с опцией *Begins With*.
2. Используйте *Property Assignment Tool* для доступа к форме пользовательских свойств.

PINNAME=NET

где PINNAME — это имя скрытого вывода, а NET — это имя сети, к которой вы хотите его присоединить.

Редактирование существующих устройств

Любой элемент устройства, собственно ли, устройства с единственным элементом, или содержимое много-элементного устройства, могут быть разбиты на составляющие, выводы и графику, при использовании команды *Decompose* из раздела *Library* основного меню.

Чтобы отредактировать графику или выводы:

1. Разместите образец устройства, как компонент.
2. Выделите объект, щёлкнув по нему.
3. Выберите команду *Decompose* из *Library* основного меню. Этим устройство будет разбито на плоскую графику, выводы и, возможно, маркер *Origin*. Вы также получите текстовый скрипт, который содержит имя устройства, префикс, упаковку и любые предопределённые свойства компонента.
4. Добавьте, удалите или отредактируйте графику, выводы и маркеры, как это требуется.
5. Реконструируйте устройство, используя команду *Make Device*, как ранее описывалось. Если вы выделите скрипт, сгенерированный командой *Decompose*, как и графику, до создания устройства, тогда вы можете избежать нового ввода свойств устройства.

Если компонент используется в текущем чертеже, вы получите приглашение с запросом, произвести ли полное обновление (*Design Global Update*)? Если вы выберете обновление, тогда все компоненты на чертеже, использующие это устройство, будут обновлены с помощью того же механизма, что и описан для команды *Pick*. Заметьте, что свойства уже размещённых компонентов не будут модифицированы, поскольку ISIS не знает, какие из свойств были изменены вручную.

Чтобы отредактировать свойства устройства:

1. Укажите, поместите и выделите образец устройства. Нет нужды его разбирать (*decompose*).

2. Вызовите команду *Make Device* и щёлкните **Next**, чтобы перейти к странице Component Properties. Будут отображены все существующие определения свойств.
3. Отредактируйте свойства, как требуется.
4. Щёлкните **Next** дважды и затем сохраните устройство в библиотеке, щёлкнув **OK**.

Как и с элементами устройства, вам будет предложено обновить компоненты в текущем чертеже.

Чтобы отредактировать упаковку устройства:

1. Укажите, поместите и выделите образец устройства. Нет нужды его разбирать (decompose).
2. Вызовите команду *Packaging Tool* из раздела *Library* основного меню.
3. Модифицируйте упаковку, как требуется.
4. Щёлкните по **Assign Package(s)**, чтобы сохранить изменения в библиотеке устройств.

Как и с элементами устройства, вам будет предложено обновить компоненты в текущем чертеже.

Два важных замечания:

- Помните, что все файлы проекта идут с их собственными копиями элементов библиотеки, которые они используют. Поэтому, изменение элемента в библиотеке не вызовут прямых изменений любого проекта, который использует эти элементы. Чтобы изменения вступили в силу, вы должны загрузить чертёж в ISIS и использовать команду *Pick* из раздела *Library* основного меню, что и произведёт операцию замены.
- Когда ISIS установлен, наши собственные библиотеки установлены в режим «только чтение». Это защищает их от ваших изменений — акция полезная для вас при нашем обновлении. Следовательно, если вы намерены изменить любой из элементов библиотеки, вы должны сохранять их в USERDVC.LIB или любой другой вашей собственной библиотеке.

МНОГОСТРАНИЧНЫЕ ПРОЕКТЫ

Введение

В очень больших проектах или при создании некоторой структуры в небольших обычной практикой является чертить разные узлы проекта на отдельных листах. Соединения между листами впоследствии организуются средствами общих имён цепей. Например, если две цепи (nets) на разных листах обозначены как MREQ, тогда они подразумеваются соединёнными (см. раздел «Имена сетей»).

Команды раздела *Design*

ISIS поддерживает многостраничные проекты и сохраняет все листы проекта в одном файле. Три команды раздела *Design* основного меню дают вам все необходимое:

- *New Sheet* — создаёт новый первый лист и загружает его.
- *Goto Sheet* — предоставляет меню листов, позволяя вам перемещаться по проекту. Для иерархических проектов выбор представлен полным деревом и вы можете переместиться незамедлительно к любому листу проекта.
- *Remove Sheet* — удаляет текущий лист. Вы можете удалить только корневые листы, но не можете удалять последний корневой лист.

Заголовки, представленные командой *Goto Sheet*, берутся из поля Sheet Title диалога *Edit Sheet Properties* или ещё из *Sheet Name*, если заголовок листа не был задан.

Заголовки листов, присвоенные им, также определяют их порядковое расположение в проекте, то есть, порядок, в котором они будут печататься, если вы все выводите на печать. Новые корневые листы начинаются с имён ROOT10, ROOT20 и т.д., и это даёт вам место для вставки ROOT15 или похожего листа, если вам понадобится вставить лист между существующими.

ИЕРАРХИЧЕСКИЕ ПРОЕКТЫ

Введение

Иерархический проект — это такой, который состоит из листов двух или более уровней. Высший уровень более всего похож на блок-диаграмму, показывающую структуру всей системы, а каждый блок будет иметь подсхему с частью схемы на нем. В зависимости от сложности проекта эти подсхемы могут сами содержать «чёрные ящики» или модули; ISIS не ограничивает глубину иерархии, хотя вы делаете что-то необычное, если вам нужно более, чем полдюжины уровней.

Второе использование вопроса иерархии — повторение части проекта; простым примером

может служить стерео усилитель, который имеет два моно канала и общее питание. Нет причин

останавливаться, просто рисуйте один канал, экспортируйте его как SEC-файл, и затем импортируйте его на второй лист. Однако, если вы захотите что-то изменить, даже если это косметические изменения, вы должны изменить это на чертежах для обоих каналов. Когда имеет быть более двух копий схемы, это может вызывать серьёзные трудности. Применяя иерархию, вы имеете два модуля, маркированные LEFT и RIGHT, но каждый ассоциирован с теми же данными цепи. Естественно, вам понадобятся разные ссылки для одинаковых компонентов в каждом образце моно усилителя и это обеспечивается значениями *Design Global Annotation*.

В ISIS иерархия также помогает при создании и использовании параметризованных цепей (Parameterized Circuits), и важна при разработке моделей симуляции для VSM. Это подробнее описано в разделе «Параметризованные цепи», а некоторые детали можно найти в руководстве к VSM.

Терминология

Прежде, чем мы погрязнем в дальнейших довольно абстрактных концепциях, нам нужно определить некоторые термины...

Схема (circuit)

Схема — это набор компонентов и других объектов, в общем случае соединённых. Например, мы говорили о схеме монофонического усилителя.

Лист (sheet)

Лист — это образец схемы и имеет уникальный набор аннотированных данных, которые принадлежат компонентам схемы. Когда лист связан с модулем на следующем верхнем уровне, родительском листе, мы можем назвать его подсхемой или производным листом. Однако мы можем сказать, что левый и правый каналы нашего усилителя начерчены на левом и правом «подлистах». Листы верхнего уровня проекта называются корневыми листами.

Модуль

Модуль — это объект, который имеет связанный с ним «подлист». Есть два типа модулей: подсхема и модуль компонентов.

Свойства листа

Свойства листа — это задание свойств, которые закреплены за отдельным листом и доступны для использования в выражениях свойств для любого из объектов на листе. В иерархических проектах любые пользовательские свойства родительского модуля становятся свойствами листа для производного листа.

Параметризованные цепи

Параметризованные цепи — это такие, в которых одно (или более) значение компонента или свойство другого объекта задаётся, как выражение свойств, производящее одно или более

свойств листа. Как только эти свойства листа могут быть заданы в родительском модуле (будет это подсхема или модуль-компонент), получается так, что схема сама может иметь разные значения компонента или свойств от одного образца к другому. Типичное приложение этого — цепи фильтров, в которых значения некоторых резисторов и конденсаторов разные для разных образцов схемы.

Подсхемы

Редактирование подсхемы обычным образом позволяет вам ввести ссылку, имя схемы и, возможно, некоторые пользовательские свойства, которые станут свойствами листа на производном листе. Ссылка будет LEFT или RIGHT в нашем примере усилителя, а имя схемы будет AMP.

Соединение между родительским листом и подлистом создаётся по значению того, как названы порты модуля на левом и правом краях подсхемы и выводы на производном листе. Подсхемы особенно удобны в случае, когда внешний интерфейс между родительским и производным листами не вполне ясен вначале — вы можете легко добавить или удалить порты и выводы (terminals).

Чтобы задать иерархию в подсхеме:

1. Выберите иконку *Sub-Circuit Mode* и создайте рамку для подсхемы, используя левую клавишу мышки.
2. Из *окна выбора* выберите и поместите подходящие типы портов модуля на левой и правой стороне рамки подсхемы. Вам понадобится по одному порту для каждого соединения между родительским и производным листом. Для соединения, обычно, один вход слева и один выход справа.
3. Либо непосредственно, либо используя *Property Assignment Tool*, задайте имена портов модуля. Эти имена должны соответствовать *Logical Terminals* (логическим контактам), которые вы поместите на производном листе.
4. Отредактируйте саму подсхему и дайте имена образцу (то есть, LEFT) и схеме (то есть, AMP). Некоторые подсхемы могут обобщать одно имя схемы, но на данном листе каждая должна иметь уникальное имя образца.
5. Остановитесь на подсхеме и нажмите **CTRL+C**. Это заставит ISIS загрузить производный лист. Пока вы не задали имя уже существующей схемы, вы должны увидеть чистый лист.
6. Выберите *Terminal* (из подменю, вызванного щелчком правой клавиши мышки) и поместите выводы для соответствия портам модуля на подсхеме.
7. Вновь либо непосредственно, либо с помощью *Property Assignment Tool* пометьте их соответствующими именами, соотнесёнными с именами портов модуля. Компилятор netlist создаст предупреждение для любого порта модуля, который не соответствует выводу (terminal).

8. Нарисуйте схему на производном листе, соединив с нужными выводами.

Модуль-компоненты

Любой обычный компонент может быть превращён в модуль установкой флажка *Attach Hierarchy Module* в диалоге *Edit Component*. Значением компонента станет имя ассоциированной схемы, а ссылка на компонент обслуживается как имя образца.

Соединение между родительским и производным объектами реализуется через выводы (terminals) производного, соответствующие именам выводов (pin) родительского модуля-компонента. Это работает и для скрытых выводов питания, хотя это не так, если выбрана опция *Global Power Nets* в *Edit Design Properties*.

Модули-компоненты более всего подходят для поддержки компонентов, которые нуждаются в расширении каким-то образом для целей симуляции, но сохраняются как компоненты при разводке печатной платы. *Depth control* (контроль глубины) по команде на форме *Netlist Generator* (раздел этого руководства *Depth*) предлагает возможность выбора того, что случится.

Чтобы установить иерархию модуль-компонента:

1. Выберите и поместите сам компонент обычным образом.
2. Вызовите диалог *Edit Component* и установите флажок *Attach Hierarchy Module*. Также убедитесь, что ссылка компонента и значение подходят в качестве имени образца и имени схемы соответственно. Как правило, модуль-компонент будет IC какого-то вида, так что его значение будет восприниматься как имя прикреплённой схемы.
3. Укажите на компонент и нажмите **CTRL+C**. Это приведёт к тому, что ISIS загрузит производный лист. Пока вы не задали имя уже существующей схемы, вы должны увидеть чистый лист.
4. Выберите *Terminal* (из подменю, вызванного щелчком правой клавиши мышки) и поместите выводы для соответствия выводам родительского компонента.
5. Либо непосредственно, либо используя *Property Assignment Tool*, задайте имена выводов (terminals), чтобы было соответствие между ними и выводами (pins) компонента. Если они не известны и не появились на экране, имена выводов (pins) можно установить, указав на конец вывода и нажав клавишу **CTRL+E**. Предупреждения компилятора netlist генерируются для любого контакта (pin) не соответствующего выводу (terminal).
6. Нарисуйте схему на производном листе, соедините её с нужными выводами (terminal).

Внешние модули

После создания модуля-компонента и производного листа со схемой можно сохранить производную схему вне проекта, так чтобы можно было использовать её в других проектах. Более того, вы можете модифицировать элемент библиотеки для родительского компонента так, что вновь размещённые образцы будут автоматически связаны с производным модулем.

Чтобы установить внешний модуль в связке с элементом библиотеки:

1. Установите иерархию модуля-компонента, как описано ранее.
2. Перейдите на производный лист, выполните команду *Edit Sheet Properties* из раздела *Design* основного меню и установите флажок *External .MOD File*. При этом создаётся MOD-файл с тем же именем, что и элемент библиотеки, ассоциированный с родительским компонентом.

В первый раз этот файл будет создан в той же директории, что и файл проекта.

3. Вернитесь к родительскому листу, выделите родительский компонент и выполните команду *Make Device*.
4. Введите имя файла модуля в поле *External Module* (внешний модуль).
5. Щёлкайте по **Next**, пока не окажетесь на странице *Library Selector*, здесь щёлкните по **ОК**, чтобы сохранить устройство в его библиотеке.

Вновь размещаемые образцы устройства будут автоматически прикреплены к MOD-файлу.

Перемещение по иерархическому проекту

Есть два способа перемещаться по иерархии:

- Команда *Goto Sheet* отобразит полную иерархию проекта в графическом виде, и вы сможете перемещаться непосредственно к любому листу проекта.
- Команды *Next Sheet* и *Previous Sheet* в разделе *Design* основного меню позволят вам сделать один шаг вниз или вверх по проекту. При использовании горячих клавиш **CTRL+C** укажите (только поместите курсор на объект) модуль, чей лист вам нужен.

Если вы «вошли» в модуль, у которого пока ещё нет имени схемы, новое внутреннее будет выбрано автоматически. Вы можете отредактировать имя схемы позже, но это отсоединит старую схему от модуля, но не изменит имени схемы. Любые схемы, которые «осиротели» из-за этого, останутся в файле проекта, но могут быть удалены с помощью команды *Tidy* из раздела *Edit* основного меню.

Глобальные аннотации проекта

Когда несколько объектов подсхем обобщают одно имя схемы, вы можете обнаружить, что изменения при редактировании одного из них отражаются на всех них. Это следует из факта, что вам нужно каждый тип схемы отрисовать только однажды. Однако каждый образец имеет свой

собственный набор ссылок на компоненты для объектов схемы. Это, конечно, необходимо для разводки печатной платы, где каждый образец потребует отдельного физического воплощения. Автоматический аннотатор последовательно поддерживает Design Global Annotation (глобальные аннотации проекта), а вам нужна только уверенность в том, что изменения ссылок компонентов на одном из подлистов не скажутся на ссылках других образцов схемы. Можно и запретить эту возможность для приложений специалистов — см. «Не физические листы» ниже.

Не физические листы

В некоторых приложениях, где есть множество образцов схемы, предпочтительнее для всех образцов ввести одинаковые аннотации. Если в netlist'e воспроизводятся эквивалентные части, в каждом образце должны быть различия и это достигается через «не физические листы». В нашем примере усилителя первый IC в каждом канале будет LEFT_U1 и RIGHT_U1 — эти имена будут произведены через связь с именем листа (Sheet Name, но не заглавием листа — sheet title) родительского модуля, подчёркивание и ссылку на элемент.

Не физический лист выбирается через команду *Edit Sheet Properties*, когда релевантный лист предварительно загружен. Установите флажок *Non-Physical Sheet*, если вы хотите, чтобы имена компонентов были локальными на отдельном листе.

Эта функциональность, похоже, не будет полезна, когда конечным результатом должен стать проект по разводке печатной платы, но полезна при симуляции или тогда, когда модуль представляет дочернюю карту.

Имена сети на подлистах всегда будут иметь похожий префикс, если это не сеть питания и опция *Global Power Nets* в *Edit Design Properties* выбрана. Результатом станет то, что имена сетей на подлистах будут «локальными»; это более или менее важно, когда используется несколько образцов той же схемы.

ВВЕДЕНИЕ

Схематическая диаграмма содержит два сорта информации: графическую и электрическую. Процесс генерации netlist'a состоит в извлечении электрических данных и записи их в форме, которую могут использовать другие САД программы. К сожалению нет единого стандарта для файлов netlist, а большинство производителей «делает свою собственную вещь». В подобной обстановке только международный комитет по стандартам или очень большой и успешный производитель может надеяться на исправление положения дел. Разработчик сделал EDIF, который достаточно сложен, чтобы быть виртуально бесполезным, и нет ни одного производителя достаточно крупного, чтобы ввести стандарт de facto. Подобно другим, мы решили использовать наш собственный формат и предоставить также возможность конвертирования его в некоторые другие форматы файла, наиболее используемые.

Наш формат назван SDF по Schematic Description Format (формат описания схемы). SDF разработан так, чтобы быть компактным, читаемым и исключительно лёгким для процесса — Visual BASIC'у будет приятно. SDF также задуман, как открытый формат — техническая спецификация будет предоставлена любому по его запросу.

ИМЕНА СЕТЕЙ (NET)

Итак, что такое, в сущности, netlist? Netlist — это список сетей и, прежде чем вы спросите, сеть — это группа выводов (pins), которые соединены вместе. В ISIS вывод определяется ссылкой на компонент, которому принадлежит, его типом (определённый при создании соответствующего устройства) и именем вывода или номером.

Сеть может также иметь имя, и одна из задач компилятора netlist — соединить все сети, которым были даны одинаковые имена. Соединения между группами выводов могут, таким образом, быть определены без явного проведения проводника между ними. Эта возможность полезна для того, чтобы избежать загромождения чертежа, и важна для задания соединений между листами в многостраничных проектах. В следующих случаях имя будет ассоциировано с сетью:

- При прикреплении этикетки к проводу для любого из них в сети — сеть получит имя с этикетки провода.
- При соединении с Logical Terminal — сеть получит имя от Terminal.

Если в нескольких случаях обнаруживаются разные имена, сеть возьмёт все имена и соединит с другими сетями, которые имеют любое из этих имён. Окончательный SDF netlist выберет одно имя для сети и предшествующей схемы, основываясь на типах разных доноров имён, используемых для выбора — специальная последовательность уменьшения приоритета это:

Power Rails & Hidden Power Pins (см. ниже)

Bi-Directional Terminals

Output Terminals

Input Terminals

Generic Terminals

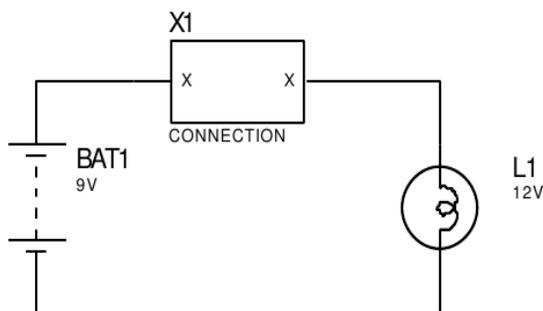
Bus Entries & Wire Labels

Как особый случай, безымянные Power Terminals получают имя VCC, а безымянные Ground Terminals — имя GND.

Имена сетей могут состоять из любых символов алфавита плюс знак минус (-) и символ подчёркивания (_). Пробелы могут использоваться в PROTEUS окружении, но могут стать причиной проблем для других программ. Символы восклицания (!) и звёздочки (*) имеют специальное назначение, о чем будет сказано далее в этом разделе. Сравнение имён сетей — контекстно чувствительно.

ДУБЛИРОВАНИЕ ИМЕН ВЫВОДОВ

Если элемент устройства имеет более одного вывода с тем же именем, эти выводы будут подразумеваться внутренне соединёнными. Например, при симуляции схемы ниже лампочка будет гореть, поскольку два X вывода подразумеваются соединёнными.



Обычно это важно для устройств, которые имеют множество выводов питания и общих, но также может быть полезно для соединения матриц на схеме, таких как клавиатура или сборка светодиодов.

СКРЫТЫЕ ВЫВОДЫ ПИТАНИЯ

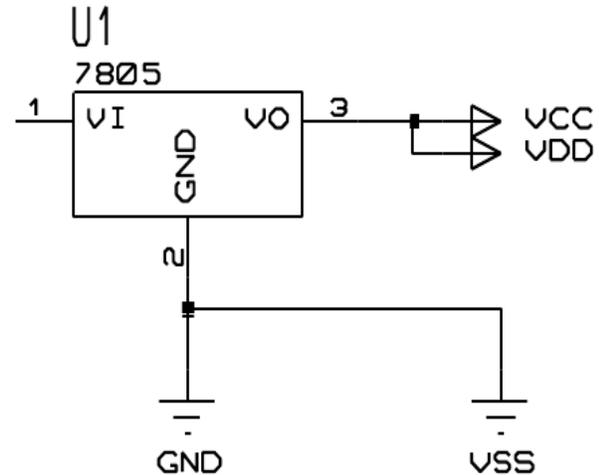
Множество интегральных схем в Device Libraries имеет скрытые выводы питания. Когда netlist генератор обнаруживает это, он создаёт новую сеть и присваивает ей имя скрытого вывода. Для вентиля 7400 будет сгенерировано две сети: VCC для вывода 14 и GND для вывода 7. Таким образом, все сети с подобными именами будут соединены, а выводы в подобных случаях соединены между собой.

Вы можете сделать дальнейшее соединение с этими сетями размещением объекта с именем-донором (то есть, Logical Terminals) с тем же именем и последующем присоединением к нему.

LABCENTER ELECTRONICS

Например, чтобы соединить резистор с VCC, вы помещаете Power Terminal без имени (он автоматически получит имя VCC), а затем проводите соединение от него к нужному выводу резистора.

В некоторых проектах, особенно там, где смешивается CMOS и TTL логика, вам может понадобиться соединение двух групп скрытых выводов питания вместе — VCC и VDD/GND и VSS, например. Это возможно при размещении двух Generic Terminals, соединении их вместе и маркировке их именами по соединению. Подходящим местом для этого часто оказывается выход PSU цепи — выход регулятора может соединяться с несколькими выводами.



В некоторых случаях вам может понадобиться соединить скрытые выводы с разными сетями по именам выводов. Это можно выполнить добавлением подходящих по имени пользовательских свойств для элементов, содержащих скрытые выводы питания. Например, если добавить элементу 7400 свойство:

$$VCC = VCC1$$

то вывод 14 будет соединён с VCC1. Заметьте, что в случае многоэлементных компонентов, как 7400, вы должны добавить свойство для всех вентиляей.

- Вы можете увидеть и редактировать имена сетей, присвоенные скрытым выводам компонента щелчком по кнопке **Hidden Pins** диалога *Edit Component*.
- Заметьте, что если поместить несколько выводов с одинаковыми именами, и только некоторые из них скрытые, то они будут все соединены вместе, но имя сети не будет сгенерировано. Вместо этого они будут соединены с проводами, подходящими к видимым выводам.

СИНТАКСИС СПЕЦИАЛЬНЫХ ЦЕПЕЙ

Глобальные сети

Подчас в иерархических проектах полезно иметь возможность сделать соединение для производного листа непосредственно с другим листом (родительским или производным) без «беготни» вверх и вниз по иерархии. Типичный случай — либо при отладке проекта в VSM, либо при использовании таких сигналов, как тактовая частота, общая для всего проекта. В любом случае ISIS распознает восклицательный знак (!) перед именем сети, как определение

глобальной цепи. Так что, вывод маркированный как !CLK, будет определяться, как требующий соединения с любым другим, маркированным как !CLK, и также с любым выводом, маркированным только как CLK на первом листе.

Заметьте также:

- Нет необходимости делать это для сетей питания, пока у вас не установлена опция *Global Power Nets* в диалоге *Edit Design Properties*.
- Безымянные выводы питания и земли фактически имеют имена !VCC и !GND, будучи глобальными сетями, пока вы не переименуете их.

Внутриэлементные соединения для многоэлементных компонентов

Эта возможность была создана специально для устранения неясных проблем при создании VSM моделей, но никак не для общего использования. Однако мы документируем её здесь для полноты руководства.

Рассмотрим модель сдвоенного операционного усилителя 1458. В сущности требуется два образца этой модели и они обобщают некоторые выводы питания. Однако устройство 1458 имеет только выводы питания, изображённые на элементе А. Как затем задать питание для элемента В?

Мы решаем это декларированием того, что имя вывода сети *V+ на производном листе задаёт внутреннее соединение между сетями всех производных листов, прикреплённых к тому же родительскому элементу, и между этими цепями и любыми выводами V+ на родительских элементах компонента. Триггер этого механизма — это предшествующий знак звёздочки (*).

ПРАВИЛА СВЯЗНОСТИ ШИН

ISIS поддерживает выводы шины и соединение между выводами шины. В основном, это работает интуитивно, но необходимо остерегаться поведения ISIS в некоторых наиболее тонких ситуациях, которые могут появиться при использовании этой возможности.

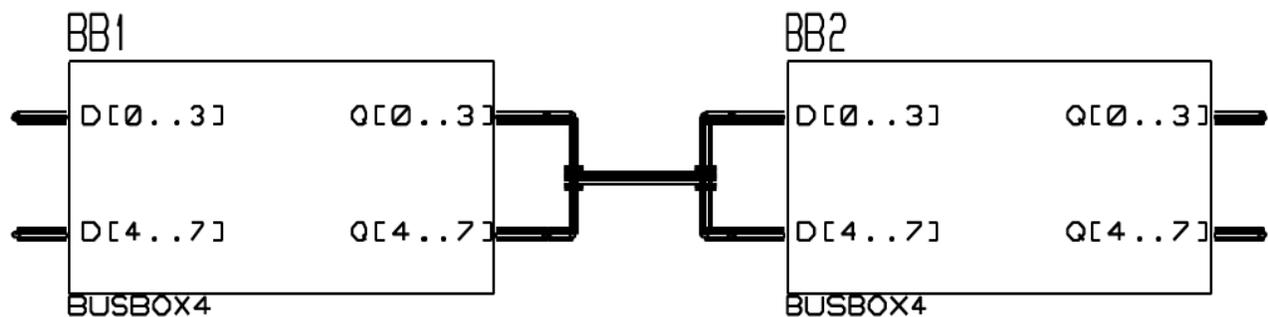
Пользователи ISIS 3.0X получили примечание, что мы изменили поведение связности шины в версии 3.1X с целью сделать его более интуитивным и не склонным к ошибкам.

Правило расположения базы

Для компилятора netlist все входы шины (выводы, контакты и порты модуля) присваиваются разряду шины. Это поддерживается внутренне в терминах базы и её ширины, так что шина D[0..7] имеет базу 0 и ширину 8.

Фундаментальная основа для связности шины в ISIS — это то, что все входы в шину (исключение для этикеток шин около точек соединения) соединяются с помощью *Base Alignment*. Это означает, например, что две шины D[0..3] и Q[4..7] соединены безымянной шиной так, что D0 будет соединено с Q4, D1 с Q5 и т.д.

Правило расположения базы применимо, даже если выводы шины были соединены с разными сегментами той же шины. Например, диаграмма ниже интерпретируется как 4x битовая шина, которая соединяет Q0 с Q4, с D0, с D4 и Q1 с Q5, с D1, с D5 и т.д.

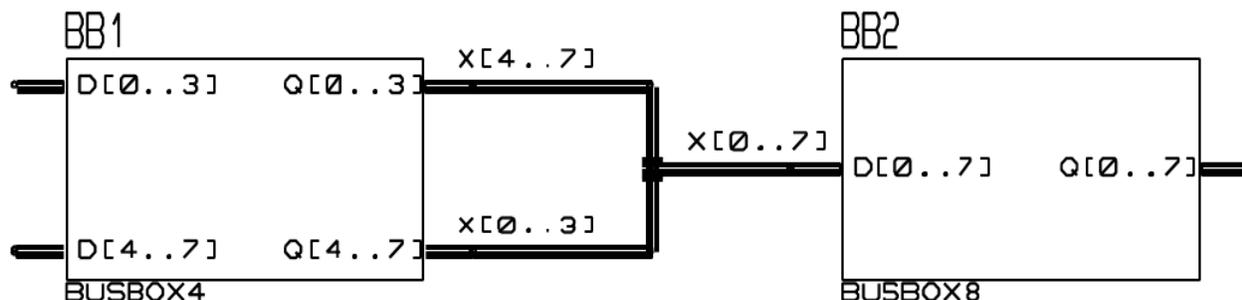


Если это не то, что требуется, тогда вы должны использовать маркировки шин (bus labels), чтобы определить требуемую связность, как это поясняется в следующем разделе.

Использование этикеток шин для изменения правила связности

Единственное исключение из правила расположения базы — это в ситуации, где несколько маркированных секций шины комбинируются в точке соединения шины. В этом случае секции шины комбинируются на основе Like Bit.

Пример ниже показывает, как некоторые выводы шины могут быть скроссированы с помощью этикеток шины:

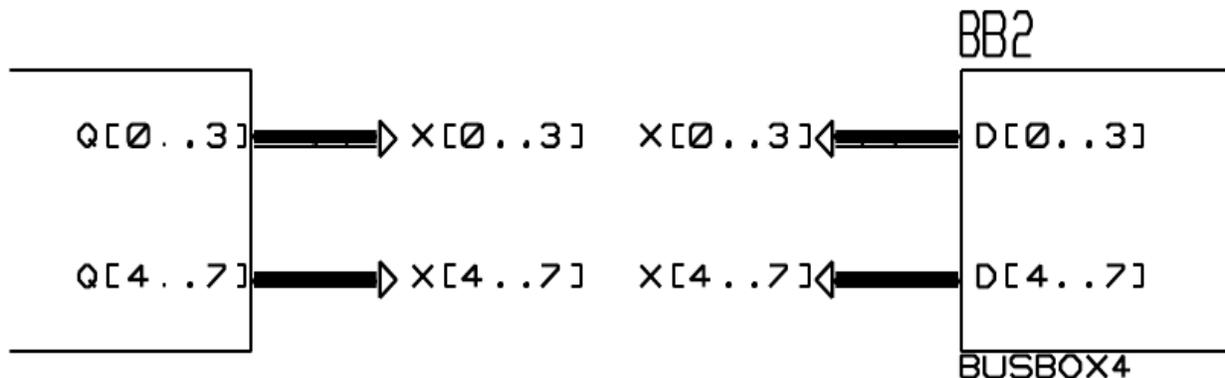


В этом случае Q0 соединено с D4, Q1 с D5, Q4 с D0, Q5 с D1 и т.д. И значимость подчеркнутого здесь в том, что выбор основы имени для маркировки шин полностью не связан с именами выводов шин — вы могли использовать D[0..3] и т.д., но это никак не сказывается на связности.

Важно ещё раз подчеркнуть, что Base Alignment Rule (правило расположения базы) использует во всех случаях этикетки баз в точках соединения, так что связь между Q[0..3] и X[4..7] соединяет Q0 с X4 и т.д.

Использование контактов шины для внутренних шин

Как и с обычными проводами, есть возможность соединить секции шины без реального вычерчивания проводов шины. Это может быть реализовано с помощью этикеток шины и/или контактов шины, как показано ниже:

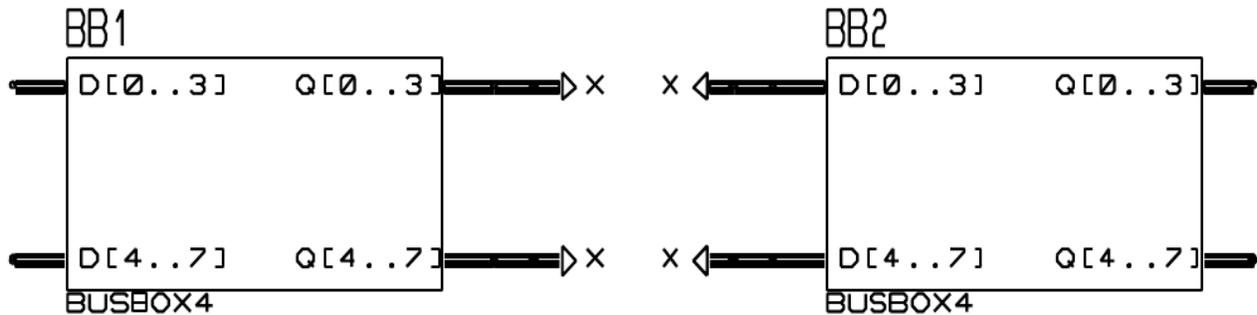


Если вы опустили спецификатор размера контакта шины или этикетки, тогда размер будет получен от секции шины, к которой вы присоединились. Размер шины определяется следующим образом:

- Если на секции шины есть этикетки, тогда они комбинируются на манер бит базиса, так что при соединении X[0..3] и X[4..7] в точке будет создан размер в этой точке X[0..7]. X[4..7] соединившись с X[8..11] создаст комбинированный размер X[4..11].
- Если нет этикеток шины на секции, тогда база размера будет 0 (позже выводы располагаются на базе) и ширина определяется самым широким контактом. Чтобы это имело место в других случаях, безымянная секция шины всегда определяется, как

имеющая базу 0, безотносительно размера контактов, которые к ней подключены.

В последнем есть ловушка для неосмотрительных. Рассмотрим диаграмму ниже:



Поскольку все контакты X имеют размер X[0..3], диаграмма, фактически, соединяет все четыре контакта шины вместе в 4х битовую шину, вместо того, чтобы создавать 8и битовую шину между Q и D. Мораль «сей басни» такова, что следует использовать исключительно размерные этикетки контактов шин во всех, даже и в простейших, случаях, если у вас есть какие-то сомнения.

Заметьте, что изолированные секции шины, в которых нет контактов, и которые не имеют этикеток или контактов выводящих размер шины, непозволительно, поскольку ISIS не может определить имена и номера индивидуальных битов для внутреннего соединения. Вместо этого, вы должны использовать схему, как показано ниже:



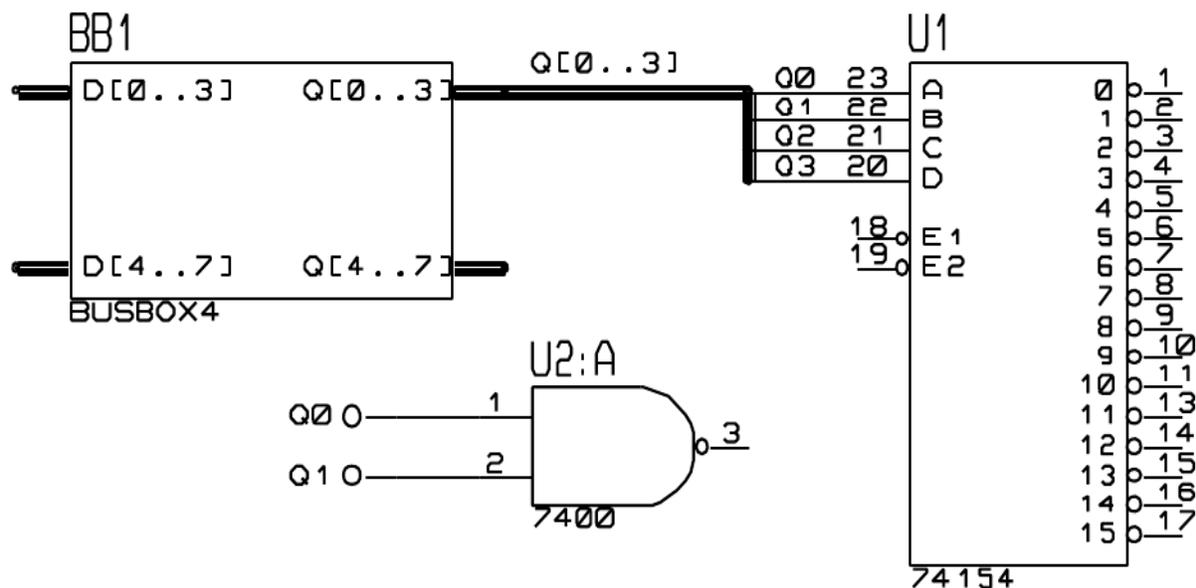
Пренебрежением этим может привести к ошибке компилятора netlist.

Соединение с индивидуальными битами

В большинстве схем, даже там, где все основные IC используют выводы шины, существует необходимость в соединении индивидуальных битов шины. Чтобы это сделать, вам нужно знать имя цепи, которое генерирует ISIS, когда вычисляет этикетку шины или контакта. С этого момента вы должны понимать, когда компилятор вычисляет обычный Logical Terminal или этикетку провода (Wire Label), объект присваивает имя сети для частичной сети. Все частичные сети, которые имеют одно или более сетевых имён, в общем, воспринимаются как требующие подключения.

Когда имя шины или контакта вычисляется, генерируется набор имён сетей, которые присваиваются частичным сетям, что составляют каждый бит шины. Этикетка шины D[0..7], следовательно, генерирует имена цепей D0, D1 ... D7.

В схеме ниже и вентиль NAND, и 74154 соединены с Q[0..3] этим механизмом.



Есть несколько моментов, которые следует отметить в этом примере:

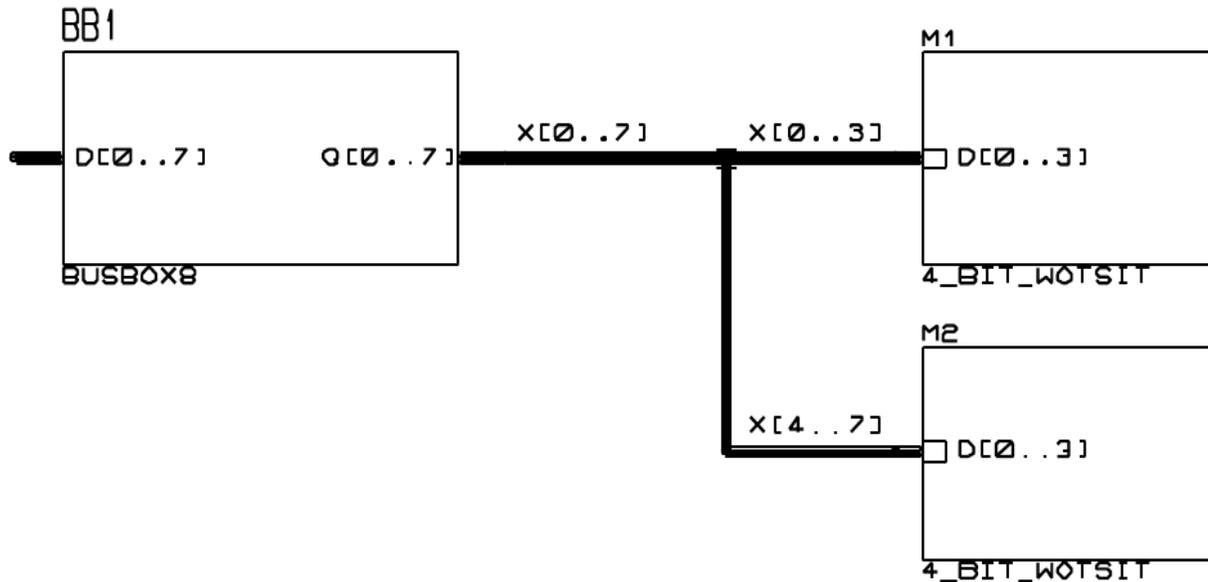
- Соединение с битами этой шины может быть выполнено без реальной отрисовки проводов к шине. Этикетка шины Q[0..3] даёт имена цепям Q0-Q3 по битам шины, и это может быть ссылкой и соединением на обычный манер.
- Этикетка шины Q[0..3] обязательна. Выводы шины Q[0..3] не вносят вклад в любые имена цепей, поскольку это может привести к нежелательным связям, если есть несколько похожих имён контактов в проекте. Такое поведение также совместимо с поведением обычных, не шинных выводов, которым не присваиваются имена цепей.

Фактически, соединение между проводом и шиной не имеет значения, пока им не «озаботится» ISIS.

- В схеме выше вы можете только маркировать шину как Q, поскольку вывод шины Q[0..3] будет обеспечивать информацию о размере.

Ветвление больших шин

Не совсем общая ситуация — возникла необходимость для шины в разбивке большой шины на несколько меньших, которые соединяются с микросхемами, возможно, 4x или 8и битовой ширины.



Здесь 8-битовый выход Q от BUSBOX8 разветвляется на две четырехбитовых шины, которые затем входят в 4_BIT_WOTSIT модули подсхемы. Base Alignment Rule применяется к соединению X[4..7] с D[0..3], чтобы получился желаемый результат. Этикетка X[0..7] в действительности избыточна в этом случае, но она и не приносит вреда, но вносит ясность.

Эта диаграмма также показывает, как связность шины может быть скомбинирована с иерархическим проектом, чтобы получать сложные схемы для представления массивов схем. Порты модуля ведут себя также, как выводы для целей связности шины.

Общие комментарии и предупреждения

Предыдущие разделы описывали поведение выводов шины и этикеток в контексте осмысления их использования. Вместе с тем, можно нарисовать другие (неординарные) возможности, и есть надежда, что объяснение того, как выводы шины работают, даст вам повод поразмышлять, что из этого получится.

Однако, если вы в чем-то сомневаетесь, например, в результирующей связности того, что вы изобразили, мы очень рекомендуем, чтобы вы проверяли результирующий netlist в текстовом редакторе до того, как просто пренебречь проверкой, считая, что связность окажется такой, какой вы её задумывали.

Строго говоря, вам не так просто будет ошибиться, если вы запомните следующее:

- Base Alignment Rule применимо всегда, исключая случай, когда маркированные секции шины сливаются в точке соединения с шиной.
- Используйте не имеющие размера контакты шины только в простейших случаях; не маркированные контакты шины или порты модуля получают базу 0, если нет других меток на этой секции шины.

ГЕНЕРАЦИЯ ФАЙЛА NETLIST

Команда *Netlist Compiler* в разделе *Tools* основного меню вначале представляет диалоговую форму *Netlist Compiler*, а затем селектор файлов, в котором вы выбираете имя файла для netlist. В большинстве случаев достаточно установок по умолчанию; при этом генерируется физический netlist в SDF формате для всех листов проекта. Функции различного управления следующие:

Формат

Может быть сгенерирован netlist в разных форматах — SDF формат Labcenter Native, а другие форматы используются для совместимости с программами других производителей. Некоторые беглые замечания представлены в разделе «Форматы Netlist'a», тогда как детальное описание приложений, касающихся некоторых корпусов, доступно через техническую поддержку нашего отдела. Поскольку программы других производителей — это предмет изменений вне нашего контроля, предпочтительнее обращаться к примечаниям приложений, а не к печатным руководствам.

Логический/физический/перенос

Логический netlist содержит имена выводов, тогда как физический netlist содержит номера выводов. Более тонкое различие в том, что в физическом netlist'e элементы многоэлементного компонента, такого как 7400, будут сгруппированы (появляясь, например, как U1), тогда как в логическом netlist'e они сохраняются разделёнными (появляясь, например, как U1:A, U1:B, U1:C, U1:D). Логический netlist будет в основном использован для симуляции, а физический больше подходит для разводки печатной платы.

Режим переноса используется только в приложениях специалистов ISIS, для которых предоставляется отдельная документация.

Границы

Предопределённые границы — это весь проект. Текущие границы ограничивают генератор netlist'a до листа, который загружен. Это обычно полезно, когда вы хотите извлечь netlist из производного листа, возможно, дочерней карты, которая должна быть представлена при работе с ARES, но является частью большого проекта, который требуется симулировать полностью. Также возможно создать «virtual test jig — шаблон виртуального теста», имея родительский лист, который содержит схему для симуляции компонента на производном листе.

Глубина

Режим по умолчанию — это выравнивание проекта. В этом случае объекты с подлистами будут заменены их реализацией. Если netlist не выровнен, тогда это замещение не произойдёт, и объекты подлистов появятся «как есть» в partslist и netlist.

Наиболее общие соображения для того, чтобы не выравнивать проект — это если некоторые

компоненты имеют привязанные к ним производные листы, представляющие их модели для симуляции, но вы хотите произвести чисто физический netlist для разводки печатной платы. Заметьте, что если вы хотите использовать эту возможность, вы не сможете получить иерархический проект — даже ISIS не пригоден для селективного выравнивания! Однако с VSM это не приговор, поскольку вы можете легко скомпоновать внешний файл модели с проектом, используя свойства MODFILE. Смотрите руководство к VSM.

Ошибки

Разного рода ошибки могут обнаружиться при генерации netlist'a; наиболее общая — когда два элемента имеют одинаковые имена. Если обнаруживаются какие-то ошибки, они отобразятся в окне всплывающего обозревателя текста.

ФОРМАТЫ NETLIST'a

SDF

Schematic Description Format (формат описания схемы) — это фирменный формат Labcenter — используемый VSM, ARES и любыми будущими EDA продуктами Labcenter. Он также легко читается и преобразуется в другие формы. Он содержит всю текстовую/связную информацию, включённую в файл DSN.

Используйте *Physical mode* (физический режим) для ARES.

BOARDMAKER

Netlist формат для Tsien Boardmaker II.

Используется пользовательское свойство PACKAGE для имён корпусов, если файл генерируется из диалога *Netlist Compiler*. Если вы хотите использовать другое поле, вы должны вызывать генератор netlist'a из файла скрипта.

Используйте *Physical mode* (физический режим).

EEDESIGNER

EE Designer III netlist формат.

Комментирует упаковку, как для Boardmaker.

Используйте *Physical mode*.

FUTURENET

Netlist формат, используемый инструментом разработки Dash. Также популярен для общих

целей переноса netlist'a.

Используйте *Physical mode* для списка выводов (Pin List), логического режима (Logical mode) для Net List.

MULTIWIRE

Формат Multiwire netlist. Также используется программой разводки плат EAGLE.

Формат файла не содержит данных корпусов.

Используйте *Physical mode*.

RACAL

RACAL netlist формат. Используется RedBoard, CADSTAR и т.д.

Комментирует упаковку, как для Boardmaker.

Создаются два файла с расширениями CPT и NET.

Используйте *Physical mode*.

SPICE

SPICE netlist формат, также идеален для P-Spice.

Цепь земли будет узлом 0, безымянные цепи начинаются с 1000; нумерованные цепи будут поддерживаться непосредственно. Файл SPICE.LXB может быть переименован в SPICE.LIB, чтобы получить набор моделей, совместимых со SPICE.

Используйте Logical format.

Не используйте этот формат для создания моделей PROSPICE — используйте нормальный MDF вывод из *Model Compiler*, он более гибкий.

SPICE-AGE FOR DOS

SPICE-AGE netlist формат для Those Engineers аналогового симулятора.

Файл SPICEAGE.LXB может быть переименован в SPICEAGE.LIB, чтобы получить набор моделей, совместимых со SPICE-AGE.

TANGO

Tango netlist формат, также используемый Protel и другими. Также хороший формат для общих целей.

Комментирует упаковку, как для Boardmaker.

LABCENTER ELECTRONICS

Используйте *Physical mode*.

VALID

Valid netlist формат, используемый для переноса проектов ISIS в VALID Transcribe упаковку.

Используйте *Transfer mode*.

VUTRAX

Netlist формат для использования с программой VUTRAX PCB design, как используется несколькими проектами bureaux.

Комментирует упаковку, как для Boardmaker.

BILL OF MATERIALS

Генерация отчёта

ISIS может генерировать Bill of Materials (спецификация материалов), который содержит список всех компонентов, используемых в текущем проекте. Вам предоставляется исчерпывающий контроль за содержанием и компоновкой этого отчёта.

Чтобы сгенерировать отчёт, все что вы должны сделать, это выбрать имя для конфигурации Bill Of Materials (BOM), которое вы хотите, в подменю *Bill of Materials* из раздела *Tools* основного меню. Отчёт появится в Text Viewer (обозреватель текста) и затем может быть сохранен или распечатан, если нужно.

Конфигурация Bill of Materials

Содержание и форматирование BOM определяется скриптом конфигурации. Один такой скрипт, Default, вы получаете с ISIS, и можете создавать, редактировать или удалять скрипты, используя команду *Set BOM Scripts* в разделе *System* основного меню. Команда вызывает диалог *Edit BOM Scripts*.

Заметьте, что изменения, сделанные в вашем скрипте конфигурации Bill Of Materials, скажутся только на текущей, загруженной копии ISIS. Если вы внесли изменения и хотите, чтобы они были доступны в дальнейшем, используйте команду *Save Preferences* из раздела *System*, чтобы сохранить скрипт в реестре.

Пример скрипта конфигурации показан ниже:

```
REFWIDTH=20  
FIELD=VALUE,15  
TOTAL=COST,10  
CATEGORY=M,Modules  
CATEGORY=R,Resistors  
CATEGORY=C,Capacitors  
CATEGORY=U,Integrated Circuits  
CATEGORY=Q,Transistors
```

Пояснения разных ключевых слов следующие:

REFWIDTH	Определяет количество колонок для ссылок компонентов.
FIELD	Задаёт свойства компонента и количество колонок, предназначенных для их отображения.
TOTAL	Задаёт свойства компонента, которые будут итоговыми и количество колонок для отображения результата.
CATEGORY	Задаёт категорию, под которой соберутся похожие компоненты. Элементы в проекте категоризованы согласно алфавитному порядку ссылок. Если нет подходящей категории, элемент попадает в категорию, названную «Miscellaneous — разное». Категории появляются в отчёте в порядке записи в файле конфигурации.

Возможности конфигурации Bill of Materials в ISIS очень гибки, и, пожалуй, они дают больше, чем аналогичные у конкурирующих продуктов. Однако их может не хватать для всех мыслимых потребностей. Если вам нужен специализированный отчёт Bill of Materials, самое лучшее — разработать некую собственную программу для чтения SDF netlist и вывода данных в требуемом для вас формате. Эта задача может быть выполнена довольно легко с небольшим знанием программного «ноу-хау». Любой диалект BASIC подойдёт для этой цели.

ИМПОРТ ДАННЫХ ASCII

Хотя это не строго касается генерации отчёта, мы решили обсудить это здесь, поскольку это весьма важно, когда рассматривается в сочетании с отчётом BOM.

Идеальный ADI таков, что хранимый максимум данных с компонентом (то есть, цены, складские коды, допуски) будет оставаться тем же для каждого типа компонента на протяжении всего вашего проекта. ADI позволяет вам задать данные для разных типов компонентов в простом ASCII файле и импортировать их все в проект единственной командой.

Исходный файл ADI может быть создан с помощью любого ASCII текстового редактора, например, MS-DOS редактора EDIT или редактора Windows NOTEPAD. Файл может содержать любое число отдельных команд, каждая из которых начинается с новой строки с ключевого слова команды, а заканчивается на новой строке ключевым словом END. Есть две команды ADI: команда IF...END и команда DATA...END.

Интерпретатор ADI запускается командой ADI и затем выбирается файл ADI с помощью селектора файлов. Команды внутри файла ADI загружаются и предварительно компилируются — любые ошибки будут записаны в журнал (log) ошибок, который автоматически отображается, если ошибки обнаружены. Затем ISIS обращается к каждой команде в ADI файле, последовательно для каждого компонента проекта.

Команда IF...END

Команда IF...END позволяет вам проверить существующие свойства каждого компонента через выражение, и, если выражение приобретает значение TRUE, тогда подкоманды внутри блока IF...END последовательно выполняются применительно к компоненту. Синтаксис наилучшим образом поясняет следующий пример:

```
IF DEVICE="CAP ELEC" AND NOT VALUE=10p
VALUE=1n,HIDEKWD
TOLERANCE,HIDE
STOCKCODE,REMOVE
END
```

Выражение будет относиться к компоненту, следующему за ключевым словом IF. Выражения состоят из одного или более термов, разделённых операторами. Каждый терм состоит из значения свойства, которое может сопровождаться знаком равенства и значением.

- Терм вычисляет только TRUE, если есть имя свойства, и в том случае, когда значение было задано, значение свойства должно соответствовать значению заданному символ-за-символом, и чувствительно к регистру.
- Операторы состоят из скобок, которые могут быть использованы для выделения подвыражений, и ключевых слов AND, OR и NOT. Операторы AND и OR выполняются слева-направо, здесь нет старшинства. Оператор AND даст TRUE, только если вычисление даёт TRUE, а терм или подвыражение следуя этому вычисляются как TRUE. Оператор OR даёт TRUE, если вычисление даёт TRUE или терм, или подвыражение, следующие за ним, вычисляются как TRUE. Оператор NOT унарный и даёт результат противоположный вычислению терма или подвыражения, которые следуют за ним.
- Специальное выражение, состоящее из единственного ключевого слова TRUE, позволяет вам воздействовать на все компоненты в проекте.

В примере выражение проверяет, что образец компонента библиотеки — это элемент CAP ELEC, и что он не имеет значения 10p. В этом случае следующие три под-команды после выражения IF и перед закрывающим ключевым словом END будут применены к компоненту. Заметьте, что поскольку проверяемый элемент имеет пробел (CAP ELEC), он заключён в двойные кавычки.

Каждая из под-команд содержит необязательное имя свойства с последующим знаком равенства и новым значением и/или запятой и командой. Если значение задано, имя свойства либо создаётся с новым значением, либо модифицируется его существующее значение. Если задана команда, она выполняется следуя любому значению, как следующие:

LABCENTER ELECTRONICS

SHOW	И имя свойства, и его значение должны быть видимы.
HIDE	И имя свойства, и его значение должны быть скрыты.
HIDEKWD	Имя свойства скрыто; видимость значения свойства не изменяется.
HIDEVAL	Значение свойства скрыто; видимость имени свойства не изменяется.
REMOVE	Имя свойства и значение удаляются из компонента.

Если команда не задана, тогда существующая видимость имени свойства и значения не меняются.

В примере свойство VALUE назначено для значения 1n и имя свойства скрыто, значение свойства TOLERANCE слева не изменяется, но пара property/value скрыта, и, наконец, свойство STOCKCODE удалено из компонента.

Команда DATA...END

Команда DATA...END позволяет множественную проверку значений против фиксированного списка имён свойств и даёт соответствие, позволяющее набору отдельных значений быть присвоенными отдельному набору фиксированных свойств.

Посмотрите следующий пример команды DATA...END:

```
DATA DEVICE + VALUE : COST+, TOLERANCE, STOCKCODE-
RES          1k      : 0.01, 1%,          100-1001
RES          1k2     : 0.01, 1%,          [REMOVE]
"CAP ELEC"   1n      : 0.03, 5%,          200-1001
SWITCH       *       : 0.25, [SKIP],       300-1001
END
```

Ключевое слово DATA с последующим именем или именами свойств для проверки отделены знаками плюс. Этот список продолжается двоеточием со списком имён свойств для присваивания, разделённых запятыми. Каждое имя свойства может (не обязательно) непосредственно сопровождаться символами плюс или минус, что отражает, хотите ли вы, чтобы имя свойства или значение для него были показаны или скрыты, соответственно; отсутствие значка показывает, что существующая видимость должна остаться.

В примере проверяются свойства DEVICE и VALUE. Получив соответствие, новые значения присваиваются COST, TOLERANCE и STOCKCODE свойствам; свойство COST и его значение будут сделаны видимыми, а свойство STOCKCODE будет скрыто.

Каждая линия между ключевым словом DATA и ключевым словом END состоит из списка значений свойств, которые вы хотите проверить (разделённые одним или несколькими пробелами), двоеточия и затем списка значений свойств, которые вы хотите присвоить, разделённые запятыми. Каждая линия выполняется последовательно; значение каждого имени свойства в списке слева от двоеточия в строке DATA проверяется по соответствию проверяемому значению на текущей строке. Если все значения свойств совпадают, тогда каждое свойство, именованное справа от двоеточия на строке DATA назначается соответствующему значению на текущей строке.

В примере первая строка проверяет свойство DEVICE для значения RES и свойства VALUE со значением 1k; если оба совпадают, тогда свойство COST приобретает значение 0.01, свойство TOLERANCE значение 1%, а свойство STOCKCODE значение 100-1001. Похожим образом третья строка проверяет свойство DEVICE для значения CAP ELEC (поскольку имя устройства содержит пробел, оно должно быть заключено в двойные кавычки) и свойство VALUE со значением 1n; если оба совпадают, тогда свойство COST получит значение 0.03, свойство TOLERANCE значение 5%, а свойство STOCKCODE значение 300-1001.

Заметьте, что имя элемента CAP ELEC заключено в двойные кавычки. Вы должны это делать для любого соответствия значений слева от двоеточия, поскольку есть разделение пробелом. Если этого не сделать, ISIS пожалуется на то, что ожидает двоеточия, поскольку предполагает, что вы хотите проверить свойство DEVICE на значение CAP, а свойство VALUE на значение ELEC; 1n оказывается неожиданностью. Аналогично, присваиваемые значения справа от двоеточия разделены запятыми, так что, если вы хотите присвоить новые значения, что имеют запятые, тогда вы должны заключить значения в двойные кавычки.

Есть две необычные особенности значений свойств, которые проверяет ADI.

- Значения могут содержать символы шаблонов: «?» и «*». Знак вопроса означает, что любой одиночный символ может появиться с значениями свойства при проверке в той позиции, где знак вопроса. Звёздочка показывает, что любое число символов может появиться в значении свойства при тестировании между набором символов слева и справа от звёздочки. В нашем примере выше мы проверяем свойство DEVICE со значением SWITCH, но выбираем не проверять свойство VALUE компонента, задавая в тестовом значении единственную звёздочку.
- Числовые значения (включая значения, которые содержат суффикс экспоненты, как «k» или «u»). Такие значения конвертируются ISIS в текстовое представление с шестью десятичными знаками, и экспонента и текстовое представление затем сравниваются символ за символом. В нашем примере компонент, размещаемый из библиотеки, элемент RES, а присваиваемые значения 1k, 1000 и 1.0k будут все сочетаться с условием проверки на первой строке.

Пока список справа от двоеточия нормально состоит из новых значений для присвоения подходящим именованным свойствам на строке DATA, вы можете вместо этого задать команду, как следующие:

- [NULL] Присваивает подходящему свойству пустую строку. Только имя свойства и следующий знак равенства появятся в текстовом блоке свойств компонента.
- [REMOVE] Удаляет свойство; имя свойства и его значение удаляются из текстового блока свойства компонента.
- [SKIP] Пропускает присвоение подходящему свойству. Это ключевое слово требуется, поскольку пустое присвоение (ничего между запятыми) вызовет появление ошибки. Любое существующее свойство в текстовом блоке свойства компонента остаётся не изменённым.

Вторая строка в примере задаёт, что свойство STOCKCODE любого компонента размещённого, как элемент RES с VALUE в 1k2, должно быть удалено. Аналогично, четвёртая строка задаёт, что не должно быть присваивания для свойства TOLERANCE для компонента из устройств SWITCH.

ПРОВЕРКА ЭЛЕКТРИЧЕСКИХ СОЕДИНЕНИЙ

(ELECTRICAL RULES CHECK)

ISIS может проверить проект на наличие простых ошибок, проверяя разные типы выводов (pin), присоединённых к каждой цепи. Очевидные примеры ошибок — это выходы, которые соединены вместе, или несколько входных выводов, соединённых вместе без источника сигнала. Контакты (terminals) также должны иметь электрический тип — входной контакт, соединённый со входным выводом, предполагает источник сигнала для него.

Генерация отчёта

Вы можете генерировать отчёт ERC командой *Electrical Rules Check* из раздела *Tools* основного меню. Результат отображается в *Text Viewer* (обозреватель текста), откуда он может быть сохранен или выведен на печать, как нужно.

Держите в памяти, что не всё, что появляется в отчёте, может быть действительно ошибками (то есть, некоторые входные выводы могут преднамеренно оставаться NC, не присоединены), и что более тонкие ошибки, как, например, неправильные значения элементов, не будут обнаружены. Тем не менее, множество нелепых ошибок может быть обнаружено на ранней стадии работы.

Сообщения об ошибках ERC

Первая часть процесса ERC выполняет компиляцию netlist, и уже на этом этапе могут появиться сообщения об ошибках и предупреждения.

Действительный процесс ERC, сам по себе, обнаруживает две базовые категории ошибок:

- Сети соединены так, что, похоже, в результате могут соперничать — например, два выхода, пытающихся переключаться в разных направлениях, могут привести к появлению очень большого тока.
- Сети соединены так, что нет источника сигнала. Сеть, содержащая только вывод INPUT, приведёт к появлению ошибки UNDRIVEN.

Обнаружение ошибок первого типа определяется следующей таблицей:

	PS	IP	OP	Ю	TS	PU	PD	PP	GT	IT	OT	BT	PR
PS	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
IP	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
OP	ok	ok	er	w n	er	er	er	ok	ok	er	ok	w n	er
Ю	ok	ok	w n	ok	ok	ok	ok	w n	ok	w n	ok	ok	w n
TS	ok	ok	er	ok	ok	ok	ok	er	ok	er	ok	ok	er
PU	ok	ok	er	ok	ok	ok	ok	w n	ok	er	ok	ok	er
PD	ok	ok	er	ok	ok	ok	ok	w n	ok	er	ok	ok	er
PP	ok	ok	ok	w n	er	w n	w n	ok	ok	ok	ok	w n	ok
GT	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
IT	ok	ok	er	w n	er	er	er	ok	ok	er	ok	w n	er
OT	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
BT	ok	ok	w n	ok	ok	ok	ok	w n	ok	w n	ok	ok	w n
PR	ok	ok	er	w n	er	er	er	ok	ok	er	ok	w n	w n

КЛЮЧИ:

Выводы

- PS: Пассивный
 IP: Входной
 OP: Выходной
 Ю: Вход/выход
 TS: Вывод с 3м состоянием
 PU: Подтяжка к питанию
 PD: Подтяжка к земле
 PP: Питание

Контакты

- GT: Простой
 IT: Входной
 OT: Выходной
 BT: Двухнаправленный
 PT: Пассивный
 PR: Шина питания

Сообщения

- ok: Нет предупреждения или ошибки
 w n: Выводится предупреждение
 er: Выводится ошибка

ВЫВОД НА ПРИНТЕР

Вывод через стандартный драйвер устройства Windows выполняется с использованием команды *Print* из раздела *File* основного меню, а выбор устройства печати может быть сделан с использованием команды *Set Printer*. Эта команда также позволяет вам получить доступ к заданию драйвера устройства через диалог *setup*. Сам диалог *Print* имеет несколько опций, которые раскрываются в контекстно чувствительной подсказке (*Help*).

ВЫВОД НА ПЛОТТЕР

Поддержка Windows перьевых плоттеров, к сожалению, очень плоха. Хотя драйверы предоставляются для HPGL и других плоттеров, реализация этих драйверов весьма приблизительна. Лучшие драйверы могут быть у обычных плоттеров, но мы не связывались с этим при разработке поддержки плоттеров в ISIS для Windows. Вместо этого мы положились на драйвер плоттера для Windows, который рисует только прямые линии, а затем ISIS сам делает все остальное.

Вывод на плоттер генерируется, как для точечного вывода на принтер, с использованием команды *Print* из меню *File*. Когда плоттер выбран в качестве устройства вывода, в диалоговой форме *Print Design* устанавливается флажок *Labcenter Plotter Driver*. Если установлен этот флажок, тогда ISIS настроен только на возможность драйвера для плоттера рисовать линии, линеаризованные ISIS дуги, окружности и т.п., и воспроизведение текста векторными шрифтами. Если флажок не установлен, ISIS обращается к драйверу плоттера, как к любому другому драйверу и надеется обнаружить поддержку для него во всех вызовах Windows GDI (графический интерфейс устройств), включая преобразования функций, которые плоттер не поддерживает (например, кривые Безье).

Цвета перьевого плоттера

Драйверы плоттеров способны генерировать многоцветный чертёж, согласно цветам, выбранным командой *Set Colours*.

К сожалению, драйверы плоттеров Windows не (мы не смогли найти этого) поддерживают прямого выбора перьев плоттера по номерам. По этой причине оставляем вам поиск того, как драйвер вашего плоттера картирует цвета по перьям и выбрать подходящие цвета в диалоге *Set Colours*.

БУФЕР ОБМЕНА И ГЕНЕРАЦИЯ ГРАФИЧЕСКИХ ФАЙЛОВ

Как и с печатью непосредственно на устройства Windows, ISIS может генерировать вывод в Windows для использования другими графическими приложениями. Вы должны выбрать генерацию этого вывода либо как Bitmap (растровый рисунок), либо как Windows Metafile, и вы можете перенести вывод в другое приложение либо через буфер обмена, либо через сохранение его в файле на диске.

Генерация растрового изображения (bitmap)

Команда *Export Graphics Bitmap* из раздела *File* создаст изображение платы и разместит его либо в буфере обмена, либо в файле на диске. Диалог имеет несколько дополнительных опций, каждая из которых имеет соответствующий раздел подсказки.

Генерация метафайла

Формат Windows Metafile имеет преимущество в том, что он реально масштабируемый (scaleable), тогда как растровое изображение (bitmap) нет. Однако не все приложения Windows (то есть, Paintbrush) могут читать метафайл (metafile).

Команда *Export Graphics Metafile* из раздела *File* создаст изображение платы и разместит его либо в буфере обмена, либо в файле на диске.

Генерация файла DXF

DXF формат может быть использован для переноса вывода в базируемые на DOS приложения CAD (лучше использовать буфер обмена и метафайл для переноса в базируемые на Windows CAD программы). Файл генерируется Labcenter, а не Windows, форматтёром вывода, и из-за этого многие, основанные на Windows чертёжные атрибуты, будут потеряны.

Наш сегодняшний опыт говорит о существенной несовместимости и расхождениях между приложениями в том, что такое правильный DXF файл. Иными словами — берём шесть приложений, поддерживающих DXF, и только 30% файла при обмене, кажется, работают! И это не противоречит факту, что наш DXF был протестирован с официальными Autodesk приложениями (AutoCAD, AutoSketch и т.д.). Для работы с Windows буфер обмена представляется много более удобной средой переноса.

Генерация файла EPS

Файл EPS — это форма файла Postscript, которая может быть внедрена в другие документы. Хотя в мире популярен DTP, для Windows базируемого DTP, лучше будет, если вы воспользуетесь для переноса графики буфером обмена и метафайлом.

На генерацию EPS файл распространяются те же опции вывода, что и для *Printer Output*.

ВВЕДЕНИЕ

ARES — это высокопроизводительная система разводки печатной платы (PCB) от Labcenter Electronics с таким же интерфейсом, как и ISIS, и с полной интеграцией в ISIS на основе netlist. Используя ISIS и ARES вместе, вы гарантировано производите печатную плату, полностью соответствующую схеме, и использование netlist'a при разводке печатной платы также избавляет вас от беспокойства по поводу деталей, связанных с цоколёвкой микросхем на плате. Вдобавок netlist, более или менее, важен при использовании автотрассировки.

В ISIS команда *Netlist to ARES* находится в разделе *Tools*, и при её использовании происходит две вещи:

- Если программа ARES не запущена, она запустится с аргументами командной строки, загружающими подходящий файл PCB и читающими netlist из ISIS.
- Если программа ARES запущена, тогда ISIS находит её и пересылает ей сообщение, которое говорит ей о необходимости прочитать новый netlist.

Пока идёт обсуждение относительно специфики ARES, пользователи других программ разводки плат найдут много интересного в следующих разделах.

УПАКОВКА (PACKAGING)

В плане того, чтобы программа ARES знала, корпус какой библиотеки следует использовать для данного компонента, эта информация должна быть введена где-то в процессе разработки. С ISIS и ARES лучший момент для этого либо тогда, когда создаётся новый элемент библиотеки, либо при редактировании схемы. Свойство, используемое для этого, не поверите, называется PACKAGE. Большинство элементов библиотеки в предлагаемых библиотеках имеют это встроенное свойство.

Упаковка по умолчанию

Большинство элементов библиотек, пришедших с ISIS, имеют подходящие корпуса для разводки печатной платы, ассоциированные с ними, и эти корпуса можно увидеть, когда вы просматриваете библиотеки. Заметьте, что некоторые элементы — в основном, модели примитивов для симуляции — не являются компонентами реального мира, а, соответственно, не имеют упаковки.

ISIS поддерживает множество корпусов для единственного символа схемы, и там, где это приемлемо, мы использовали это в применении и к «дырочному, PTH», и к поверхностному, SMT, монтажу для корпусов элементов библиотек.

Заметьте, тем не менее, что ответственность лежит полностью на вас — проверить, что используемое нами по умолчанию подходит для вашего частного приложения. Мы не можем и не будем принимать никаких обязательств за потери, связанные с применением неподходящих корпусов, даже если это произойдёт. Мы настоятельно рекомендуем

использовать прототипы до заказа большого количества плат у производителей.

Ручная упаковка

Пользовательские свойства, включающие PACKAGE, могут редактироваться разными способами. Простейший, выделить компонент, выбрать из выпадающего меню после щелчка правой клавишей мышки *Edit Properties*, и ввести новую информацию в текстовое поле *PCB Package*. Если компоненты уже получили ссылки, вы можете использовать команду *Edit* для доступа к ним по имени. Например, последовательность E, Q1, ENTER вызовет появление диалога для транзистора Q1. Когда множество компонентов имеют одинаковые корпуса, например, резисторы почти всегда упаковываются в RES40, можно использовать *Property Assignment Tool* для загрузки этого фиксированного значения в выбранное поле каждого компонента, по которому вы щёлкните. Например, для загрузки строки RES40 в поле PACKAGE группы компонентов, задайте строку PAT, как

```
PACKAGE=RES40.
```

Автоматическая упаковка

Для автоматической упаковки тех типов компонентов, которые имеют одинаковые корпуса, можно использовать *ASCII Data Import*. В качестве примера вместе с ISIS вы получили файл *DEVICE.ADI*, где показано, как присваиваются корпуса и другие свойства элементам в *DEVICE.LIB*.

Файл устроен так, что упаковка загружается в свойство PACKAGE, которое затем становится скрытым. Вы можете изменить это, редактируя заголовки ADI блоков — см. раздел «Генерация отчётов», где больше сказано про ADI.

Некоторые компоненты, как обычные и электролитические конденсаторы, имеют разные формы и размеры. Более того, мы не можем знать, какой серии конденсатор вы намерены использовать. Тем не менее, можно создавать ADI файлы, которые оперируют со значениями элементов, как и с именами библиотечных компонентов. Например, ниже извлеките корпуса конденсаторов от нашего любимого поставщика, и в то же время загрузите коды заказа для них в свойство ORDER.

Заметьте, что корпуса сделаны только для корпусов, которые не заданы с предустановленными размерами в *DEVICE.LIB*. Запуск *CAPS.ADI* перепишет предопределённые присваивания, чтобы добиться желаемого результата.

```
;CAPACITOR PACKAGING FOR TYPES REQUIRING
;OTHER THAN CAP10 / ELEC-RAD10 PACKAGES
DATA DEVICE + VALUE : PACKAGE-
CAP 1u : CAP20
"CAP ELEC" 470u : ELEC-RAD20
"CAP ELEC" 1000u : ELEC-RAD30
"CAP ELEC" 2200u : ELEC-RAD30
END
```

LABCENTER ELECTRONICS

```
;CAPACITOR ORDER CODES
;These are Ceramic, Polyester types and
;Electrolytics at 25V or better.
DATA DEVICE + VALUE : ORDERCODE-
CAP 100p : F146-447
CAP 680p : F146-457
CAP
1n : F149-100
CAP 1n5 : F149-101
CAP 3n3 : F149-103
.
.
.
"CAP ELEC" 47u : R11-0235
"CAP ELEC" 100u : R11-0245
"CAP ELEC" 220u : R11-0260
"CAP ELEC" 470u : R11-0280
END
```

В общем, это всегда будет финальной фазой ручной упаковки для необычных элементов или специальных компонентов, используемых только в этом проекте. Мы предполагаем, что вы сделаете свойство PACKAGE при ручной упаковке видимым, чтобы напомнить, что корпуса нестандартны.

Использование BOM для помощи с корпусами

В случае больших проектов становится трудно понять, будут ли все компоненты «упакованы», и какие корпуса они получают. Это особенно касается случаев, когда вы делаете свойство PACKAGE невидимым при *ASCII Data Import*.

Один из способов преодолеть это — создать файл дополнительной конфигурации *Bill of Materials*, который имеет следующие записи FIELD:

```
FIELD=VALUE,15
FIELD=PACKAGE,15
```

так что, оба значения элемента и корпуса появляются для каждого компонента. Затем *Bill of Materials* даёт вам отсортированный список всех компонентов и их корпусов, что существенно облегчит проверку.

Заметьте, что вы можете создать множество скриптов конфигурации *Bill of Materials*, используя команду *Set BOM Scripts* из раздела *System*.

Проверка упаковки (Package Verifier)

ISIS может проверить, все ли корпуса, заданные для компонентов схемы, есть в библиотеках ARES, и все ли номера выводов, заданные в библиотеке элементов ISIS, есть для выбранных корпусов.

Такая возможность весьма полезна при массовом создании элементов библиотеки, но и не менее

полезна, если вы хотите проверить, что все все корпуса правильны до создания netlist для ARES. Команда *Verify Packaging* может быть найдена в разделе *Library*.

Упаковка с ARES

Если вы пропустили информацию о корпусе для какого-то компонента на схеме, ARES попросит указать её, когда загружает netlist. Конечно, вы должны сделать выбор для всех корпусов на этой стадии. Однако выбор корпусов, выполненный таким путём, должен будет введён повторно — хотя бы для не размещённых компонентов — при каждой загрузке netlist'a в ARES.

Мы, заметьте, не рекомендуем этот метод работы.

СВОЙСТВА СЕТЕЙ И СТРАТЕГИИ ТРАССИРОВКИ

ARES ассоциирует с каждой цепью в netlist стратегию трассировки, которая определяет стили дорожек и отверстий, тип сети, слои трассировки и т.д. По определению, все сети получают стратегию SIGNAL, исключая:

- Цепи VCC и GND, которые получают стратегию POWER.
- Цепи с именами такими как D[0], которые получают стратегию BUS (шина). Этим заведуют квадратные скобки, которые очень важны. Подобный синтаксис предоставляется для использования с корпусами в схеме, которые иначе не передают свойства цепи для шины.

Для изменения предопределённой стратегии для данной сети, вам нужно прикрепить *Net Property* к одному из проводов цепи. Это выполняется размещением этикетки провода в форме:

```
STRAT=strategy_name
```

Если названная стратегия не существует, ARES создаст её и добавит к выбору стратегий, когда загрузит netlist. Определение стратегии может затем быть отредактировано, если выбрать стратегию и щёлкнуть по переключателю (toggle) выбора стратегии.

Совершенно справедливо размещения стратегии вида:

```
STRAT=BUS
```

на сегменте шины, хотя это не будет работать в случае «косметической» шины. Такая шина — это одна из тех, что соединяются не с контактом или выводом шины, и не имеют маркировки цепи шины.

Иногда полезно задать стратегию для всех сетей на данном листе. Например, вам может понадобиться, чтобы все цепи на листе питания получили стратегию POWER.

Это можно сделать с помощью размещения блока скрипта следующего вида:

```
*NETPROP  
STRAT=POWER
```

на подходящем листе проекта.

Этикетки свойств цепей проводов и шин получают приоритет перед глобальными свойствами сети листа.

ПРЯМАЯ АННОТАЦИЯ — ИНЖЕНЕРНЫЕ ИЗМЕНЕНИЯ

Термин «Инженерные изменения, Engineering Change» относится к ситуации, в которой существующая схема модифицируется, а полученный netlist перезагружается в ARES. Такое может случиться и при разработке оригинального проекта, и позже, когда Mark II версия продукта будет производиться. Система PROTEUS полностью поддерживает инженерные изменения, но важно, чтобы вы осознали, в каких программах могут появиться разные среды.

Добавление новых компонентов

Добавление новых компонентов и связанных с ними соединений в проект создаёт небольшую проблему, связанную с тем, что вы использовали *Auto-Annotator* в режиме наращивания. Если вы добавляете компоненты к схеме и полностью обновляете аннотацию схемы (так, что части микросхем существующих компонентов изменяются или обмениваются), тогда ARES в общем случае не сможет «понять» пришедший netlist.

Чтобы добавить новые компоненты в проект:

1. Разместите и соедините компоненты в ISIS обычным образом.
2. Используйте *Auto-Annotator* в режиме наращивания, чтобы дать элементам новые уникальные компонентные идентификаторы. В качестве альтернативы можете использовать ручной ввод. Ни в коем случае вы не должны менять идентификаторы уже существующих компонентов.
3. Перенесите изменения в ARES, используя команду *Netlist->ARES* из раздела *Tools* основного меню. ARES отобразит новые компоненты в *окне выбора компонентов (Component Selector)*.
4. Разместите компоненты в ARES обычным образом. ARES покажет линии ratsnest (не оформленные компоненты и связи), чтобы показать требуемые соединения для новых компонентов.
5. Проведите трассировку для соединения новых компонентов, используя автоматический трассировщик или ручную, как удобнее.

Удаление существующих компонентов

И ещё раз, это довольно однозначно выполняется. ARES выделяет компоненты, которые были удалены из netlist'a, так что вы можете увидеть, что было удалено, до реального удаления этого с печатной платы.

Чтобы удалить компоненты из проекта:

1. Удалите компоненты из схемы обычным образом. ISIS автоматически удалит любые провода к ним.
2. Перенесите изменения в ARES, используя команду *Netlist->ARES* из раздела *Tools*. ARES выделит и подсветит компоненты на печатной плате, которые больше не появятся в netlist'e.
3. Проверьте выделенные компоненты, чтобы убедиться в том, что вы действительно хотите их удалить. Для каждого компонента, который вы намерены удалить, наведите на него курсор и щёлкните правой клавишей мышки.
4. Удалите остаток дорожек для старых компонентов, используя обычные инструменты разводки-редактирования в ARES.

Изменения связности

Когда существующие соединения изменены, ARES анализирует текущую связность платы и маркирует секции дорожек, у которых нет больше соединений в netlist'e, как VOID. Это показано их появлением, как мигающие жёлтым. Соединения присутствующие в netlist, но не представленные на плате, показаны как линии ratsnest обычным образом.

Чтобы изменить связность проекта:

1. Внесите изменения в соединения и связность схемы в ISIS. Процедура изменений не проявится, пока либо не изменятся провода на чертеже, либо пока не появятся синтаксические изменения, такие как маркировка проводов или контактов.
2. Перенесите изменения в ARES, используя команду *Netlist->ARES* из раздела *Tools*. ARES отметит дорожки, которых нет теперь в netlist, как VOID, и они появятся, как мигающие желтые. Соединения, присутствующие в netlist, но ещё не сделанные на плате, появятся как новые линии ratsnest.
3. Проверьте освобождённые дорожки, чтобы убедиться, что вы действительно хотите удалить их. Если так, используйте команду *Tidy*.
4. Оформите дорожки, чтобы провести новые соединения, используя автоматическую или ручную трассировку, как вам покажется лучше.

Перемаркировка компонентов и перепакровка вентиляей

Этот раздел потенциально наиболее сбивающий с толку. Есть тенденция думать, что, если у вас обычный компонент в ISIS, скажем «U35», и вы редактируете его в ISIS, переименовав в «U34», то это просто отобразится в ARES при последующей компиляции и загрузке netlist. Этого не случится. Что такая перемаркировка означает в контексте PROTEUS — «U35» был удалён, а «U34» был добавлен. Следовательно ARES выделит «U35» и добавит новый элемент «U34» в селектор элементов.

Аналогично, два вентиля 7400, скажем, «U1:A и U2:B» перемаркированы в схеме, так что их имена изменились, а это означает для ARES изменение связности проекта, то есть, провода шедшие к выводам 1, 2, 3 теперь пойдут к выводам 4, 5, 6 и наоборот.

Ключевая концепция такова, что PROTEUS использует идентификаторы элементов в ISIS для кросс-ссылок между схемой и разводкой платы. Если вы изменили идентификаторы (Ids), тогда вы изменили связность проекта, а не маркировку.

Изменение аннотаций проекта должны быть сделаны в ARES, откуда они могут быть возвращены (перемаркированными) в ISIS — см. раздел «Перемаркировка».

ОБМЕН ВЫВОДОВ/ОБМЕН ВЕНТИЛЕЙ

Многие типы компонентов имеют взаимозаменяемые выводы — например, два возможных вывода 74138 или взаимозаменяемые элементы у 7404, где шесть инверторов. Некоторые компоненты, как 7400, имеют и то, и другое — есть четыре взаимозаменяемых вентиля, каждый из которых имеет по два идентичных входа. Этот феномен может быть использован при разводке сложной платы, может оказаться легче трассировать альтернативные выводы или вентиля, чем те, что заданы в netlist. Ясно, выполнение такой «подгонки по месту» сказывается на связности проекта, и эти изменения нуждаются в фиксации и, в конечном счёте, в корректировке схемы.

PROTEUS предоставляет весьма полную, и, возможно, непревзойдённую поддержку этого аспекта EDA проблемы. Собственно, предлагаются следующие возможности:

- Схема для задания взаимозаменяемых выводов и вентиляей в библиотеке элементов ISIS.
- Возможность маркировать выводы или вентиля в ARES и видеть графическое отображение того, какие другие выводы или вентиля могут быть взаимозаменяемы, что базируется на данных из библиотек элементов ISIS.
- Оптимизатор обмена вентиляей, который пытается найти оптимальное расположение вентиляей, основанное на поиске кратчайших возможных путей соединения из «кучи ratsnest».
- Автоматический возврат изменения аннотаций, ручных и автоматических, в ISIS.
- Механизм связывания, который оберегает изменения, сделанные и в схеме, и на плате одновременно, от появления конфликтов и двусмысленности.

Задание обмена выводов и обмена вентилях для ISIS Library Parts (библиотеки)

Следующее обсуждение предполагает детальное понимание механизма создания библиотечных элементов ISIS. Пожалуйста, обратите внимание на раздел «Возможности библиотеки».

Первое, что следует понять, это то, что обмен выводов может возникнуть и для единственного элемента, и для многоэлементного устройства, тогда как обмен вентилях требует, чтобы устройство имело много элементов, более или менее, по определению. Итак есть три случая.

Задание обмена выводов для устройства с единственным элементом

Это выполнимо при использовании свойства PINSWAP; имена выводов (не номера) для взаимозаменяемых перечисляются. Таким образом, резистор с двумя выводами, названными «1» и «2», взаимозаменяемыми, может получить свойство

PINSWAP=1,2

а для 74138, который имеет идентичные входные выводы «Е2 и Е3», выводится свойство

PINSWAP=E2,E3

Если есть более одного набора взаимозаменяемых выводов, можно использовать точку с запятой для разделения набора выводов. Например:

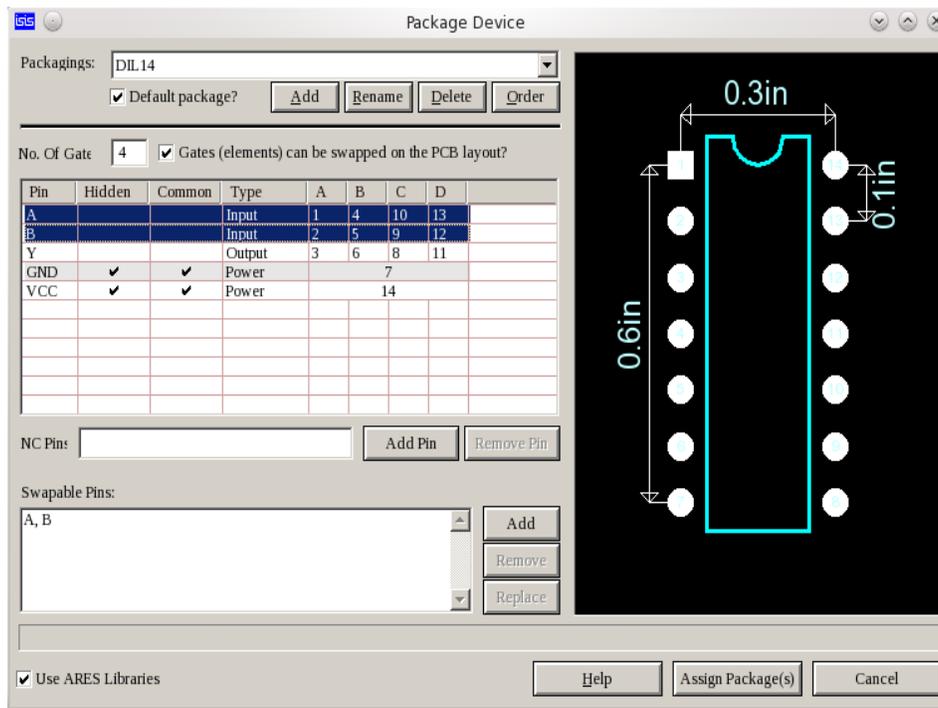
PINSWAP=A,B;C,D

означает, что А может быть обменено с В, а С с D, но обмен А-С, А-D, В-С и В-D все ещё недопустим.

Использоваться может только свойство PINSWAP.

Задание обмена выводов в многоэлементных устройствах

Для многоэлементных устройств, как 7400, показанный ниже, вы должны использовать *Visual Packaging Tool* для задания групп обмена выводов. Это выполняется с использованием раздела *Pinswap* диалога.



Чтобы добавить группу обмена выводов в упаковку:

1. Выделите выводы для обмена в Pin Grid (сетка выводов), нажав и удерживая клавишу **CTRL** и щёлкая по каждой строке последовательно. В случае выше выводы A и B взаимозаменяемы, так что вы подсвечиваете их строки, как показано.
2. Щёлкните по кнопке **Add**, чтобы создать группу pinswap.

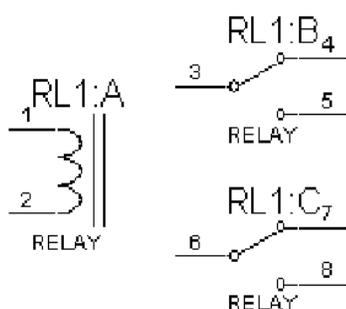
Вы можете определить множество групп обмена повторяя процедуры, описанные выше.

Задание обмена вентиляей в многоэлементном устройстве

К счастью, ISIS может выдать много информации о замене вентиляей для многоэлементных устройств и сам. Например, в случае 7400 (показано далее), скрипт упаковки содержит достаточно информации, чтобы рассказать, что взаимозаменяемые элементы — это выводы A, B и Y, и что их 4.

Все, что мы должны сообщить ISIS, чтобы позволить обмен вентиляей, это установить флажок *Gateswap* в *Visual Packaging Tool*. Заметьте, что выводы питания VCC и GND автоматически исключаются из взаимозаменяемости, поскольку они скрытые. Они также требуют прикрепления к тем же сетям, если обмен осуществляется между разными корпусами, поскольку они маркированы, как общие.

Наш завершающий пример — гетерогенный многоэлементный компонент; в этом случае DPDT реле состоит из катушки и двух пар контактов:



Pin	Hidden	Common	Type	A	B	C
C1		✓	Passive	1
C2		✓	Passive	2
COM			Passive	...	3	6
NC			Passive	...	4	7
NO			Passive	...	5	8

NC Pins:

Gates (elements) are interchangeable and can be swapped on the PCB layout?

Swapable Pins:

В этом случае выводы катушки и общие, и взаимозаменяемы. Пренебрежение тем, что они общие, конечно, будет пагубно, поскольку это позволяет обменивать контакты между разными реле!

Ручное выполнение обмена выводами и вентилями в ARES

Обмен выводов и вентиляей выполняются одинаково с использованием *Ratsnest mode* в ARES.

Чтобы обменять два вывода или вентиля:

1. Загрузите netlist для проекта из ISIS. ARES должен иметь копию netlist, которая синхронизирована со схемой до использования механизма.
2. Выберите иконку *Ratsnest Mode* в ARES.
3. Щёлкните правой клавишей по исходному выводу. Линии ratsnest, соединённые с ним, будут подсвечены.
4. Щёлкните и удержите левую клавишу мышки. Все выводы, с которыми исходный вывод может быть обменён, будут подсвечены. Этот набор включает все правильные выводы для обмена внутри исходного вентиля и все подходящие выводы в любых правильных вентилях для обмена.
5. Перетащите исходные линии ratsnest к целевому выводу, на который хотите обменять исходный. Если этот вывод в том же вентиле, тогда будет выполнен только обмен

LABCENTER ELECTRONICS

выводов. Если это другой вентиль, тогда произойдёт обмен вентилями и, возможно, и обмен выводов.

Есть ещё немного. Заметьте, что ARES не позволяет обмен там, где любой исходный или целевой вывод имеет подходящую дорожку. В случае обмена вентилями все выводы обоих вентилей должны быть не разведены (unrouted).

ПРЕДУПРЕЖДЕНИЕ

Обмен выводов и обмен вентилями производит изменение связности вашего проекта. ARES использует данные обмена, заданные в библиотеках ISIS, чтобы решить, что есть, а что нет, правильный обмен. Если есть ошибки в данных, тогда ARES может предложить неверный обмен. Мы не будем, ни при каких обстоятельствах, считать себя ответственными за любые понесённые затраты или потери, возникшие как результат таких неудач, возникли ли ошибки в вашей библиотеке или нашей, или в самой программе. Мы настоятельно рекомендуем, чтобы вы проверяли обмен, который вы сделали, на предмет его правильности, и проверили прототип платы до того, как она будет произведена в больших количествах.

Оптимизатор обмена вентилей

В тех случаях, когда есть много взаимозаменяемых вентилей, очень тяжело увидеть наилучшее расположение для них. Количество возможных комбинаций может стать астрономическим даже при небольшом количестве вентилей. Простой проект SHIFT16 позволяет много больше комбинаций, чем есть частиц во вселенной!

Чтобы помочь вам найти более оптимальные решения, в ARES встроен и автоматизирован оптимизатор, который выполняет тысячи замен дорожек, чтобы найти так называемый «технический минимум» для заданного размещения на плате. Во многих случаях это близко к оптимальному решению, и почти неизменно даёт некоторое сокращение общей длины ratsnest.

Чтобы использовать оптимизатор обмена вентилей:

1. Загрузите netlist проекта из ISIS. ARES нужно иметь копию netlist, чтобы синхронизироваться со схемой до использования этой команды.
2. Разместите все компоненты обычным образом, стремясь к тому, чтобы длина ratsnest была как можно меньше. Оптимизатор обмена не будет работать при плохом размещении!
3. Выполните команду *Gateswap Optimizer* из раздела *Tools* основного меню.

Алгоритм будет повторять проходы пока не перестанут появляться улучшения. Время выполнения может быть довольно большим (30 минут), если есть много возможных обменов.

ПРЕДУПРЕЖДЕНИЕ

Gate-Swap Optimizer использует введённые данные в библиотеках компонентов ISIS, чтобы решить, что есть, а что нет, правильно. Если есть ошибки в данных, тогда *Gate-Swap Optimizer* может внести неверные изменения в связность вашего проекта. Мы не будем, ни при каких обстоятельствах, считать себя ответственными за любые понесённые затраты или потери, возникшие как результат таких неудач, возникли ли ошибки в вашей библиотеке или нашей, или в самой программе. Мы настоятельно рекомендуем, чтобы вы использовали эту команду для прототипа платы до того, как она будет произведена в больших количествах.

ПЕРЕМАРКИРОВКА

При некоторых обстоятельствах вам может потребоваться перенумеровать компоненты на плате, по требованию производителя или сборочных нужд. Это можно сделать либо вручную, либо автоматически.

Чтобы перемаркировать вручную:

1. Убедитесь, что последняя копия netlist'a загружена в ARES.
2. Выберите правильный слой для компонента(ов), которые собираетесь маркировать.
3. Щёлкните правой клавишей мышки по этикетке компонента и выберите *Edit Properties* из выпадающего меню.

Вы получите сообщение, если попытаетесь внести изменения, которые не могут быть возвращены в схему. Соединители, выполненные с физическими контактами, наиболее подходящий пример — поскольку нет введённых в ISIS имён, и нет места для внесения изменений.

Чтобы перемаркировать автоматически:

1. Задайте любым компонентам, которые вы не хотите перенумеровывать, свойство NOANNOTATE=TRUE на схеме.
2. Убедитесь, что последняя копия netlist'a загружена в ARES.
3. Выполните команду *Component Re-Annotator* из раздела *Tools* в ARES. Этим будет выполнена перенумеровывание компонентов, основанное на их существующих префиксах и их физической позиции на плате.

Поскольку любые изменения будут отображаться на схеме, в следующий момент вы сохраните разводку печатной платы.

ВОЗВРАТ ПЕРЕМАРКИРОВКИ В ISIS

Когда сделаны обмен выводов, обмен вентилей или перемаркировка, вам потребуется перенести эти изменения обратно в ISIS, чтобы изменения были отражены на схеме, как аннотация, так и связность платы. Есть два способа выполнить это:

Полуавтоматический возврат перемаркировки

По умолчанию, вы ограждены от внесения изменений в схему, пока открыты с не сохранёнными изменениями. Меню в ISIS будет иметь текст «(Locked)», чтобы показать это положение дел.

Чтобы синхронизировать и разблокировать схему, используйте команду *Back-Annotate from ARES* из раздела *Tools* в ISIS. Эта команда вызовет сохранение изменений в ARES.

Полный автоматический возврат перемаркировки

В этом случае схема будет обновлена, когда ISIS «выйдет на сцену». Однако результатом будет то, что ARES автоматически сохранит изменения на диске.

Чтобы выбрать автоматическую операцию, установите флажок *Auto Sync/Save* в диалоге *Set Environment* раздела *System* основного меню в ISIS. Режим по умолчанию — полуавтоматический. В обоих случаях возврат перемаркировки автоматический, если изменения разведённой печатной платы были сохранены.