

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра электронной техники и технологии

**В.М. Бондарик**

## **СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ**

### **КУРС ЛЕКЦИЙ**

для студентов специальностей  
«Медицинская электроника»,  
«Электронно-оптические системы и технологии»  
дневной и заочной форм обучения

Минск БГУИР 2006

УДК 004.4 (075.8)  
ББК 32.973 я 73  
Б 81

А в т о р  
В.М. Бондарик

**Б 81**      **Бондарик В.М.**  
Системы автоматизированного проектирования. Курс лекций:  
Уч. пособие для студ. спец. «Медицинская электроника», «Электронно-оптические системы и технологии» дневной и заочной форм обуч./  
В.М. Бондарик. – Мн. БГУИР, 2006. – 272 с.: ил.

ISBN 985-444-XXX-X

Содержит теоретические материалы в виде курса лекций по курсам «Системы автоматизированного проектирования средств медицинской электроники» и «Системы автоматизированного проектирования с основами защиты информации», разработанном в соответствии с рабочей программой для специальностей I-39 02 03 «Медицинская электроника» и I-36 04 01 «Электронно-оптические системы и технологии».

Пособие предназначено для студентов специальностей «Медицинская электроника», «Электронно-оптические системы и технологии».

УДК 004.4 (075.8)  
ББК 32.973 я 73

ISBN 985-444-XXX-X

©, Бондарик В.М., 2006  
© БГУИР, 2006

## СОДЕРЖАНИЕ

ЧАСТЬ 1. ОСЕННИЙ СЕМЕСТР .....	9
1 ВВОДНАЯ ЛЕКЦИЯ.....	9
1.1 Необходимость применения САПР .....	9
Преимущества применения автоматизированного проектирования ..	10
1.2 Классификация САПР (по ГОСТ 23501.8-80) .....	15
1.3 Составные части САПР (обеспечения) (ГОСТ 22487-77).....	16
1.4 Подсистемы САПР .....	18
1.5 Основные требования и принципы создания САПР .....	18
2 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САПР .....	21
2.1 Требования к МО САПР.....	21
2.2 Методы повышения эффективности МО САПР .....	22
2.3 Основы теории графов .....	23
2.4 Матричное представление графов.....	25
2.5 Основы теории алгоритмов .....	28
Свойства алгоритмов.....	29
Эффективность алгоритмов: .....	29
2.6 Способы записи алгоритмов .....	30
3 ФОРМАЛЬНОЕ ОПИСАНИЕ КОММУТАЦИОННЫХ СХЕМ .....	33
3.1 Основные модели представления коммутационной схемы ЭА в памяти ЭВМ.....	34
3.2 Основная модель монтажного пространства .....	35
3.3 Модель электрической схемы.....	37
4 АЛГОРИТМИЗАЦИЯ ЗАДАЧ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ .....	40
4.1 Покрытие функциональных схем модулями .....	40
Алгоритм покрытия (эвристического типа):.....	41
4.2 Компоновка конструктивных элементов по коммутационным платам .....	42
4.3 Классификация алгоритмов компоновки .....	43
Последовательный алгоритм: .....	44
Итерационный алгоритм: .....	45
5 АЛГОРИТМЫ РАЗМЕЩЕНИЯ КОНСТРУКТИВНЫХ МОДУЛЕЙ .....	46

5.1	Постановка задачи размещения конструктивных модулей различных уровней иерархии .....	46
5.2	Классификация алгоритмов размещения .....	48
5.3	Непрерывно-дискретные методы размещения .....	49
	Алгоритмы, использующие градиентные методы .....	49
	Алгоритмы, использующие динамические модели .....	50
5.4	Алгоритмы, использующие дискретные методы оптимизации .....	52
	Алгоритмы назначения .....	52
	Алгоритмы случайного поиска .....	53
5.5	Эвристические алгоритмы .....	55
	Структура итерационных алгоритмов .....	56
	Алгоритмы парных перестановок .....	56
	Алгоритмы групповых перестановок .....	58
	Алгоритмы последовательной установки .....	58
	Параллельные алгоритмы на основе метода обратного размещения .....	60
5.6	Особенности алгоритмов размещения при многоцелевой оптимизации модулей .....	62
6	АЛГОРИТМЫ И МОДЕЛИ ТРАССИРОВКИ СОЕДИНЕНИЙ В ЭА .....	64
6.1	Классификация алгоритмов трассировки .....	64
6.2	Трассировки проводных соединений .....	66
	Алгоритм Краскала (Вайнберга – Лобермана) .....	68
	Алгоритм Прима .....	69
	Алгоритмы трассировки проводов в каналах .....	72
7	АЛГОРИТМЫ И МОДЕЛИ ТРАССИРОВКИ ПЕЧАТНЫХ СОЕДИНЕНИЙ В ЭА .....	74
7.1	Постановка задачи трассировки печатных соединений .....	74
7.2	Волновые алгоритмы. Волновой алгоритм Ли .....	77
	Примеры .....	78
	Проведение пути минимальной длины .....	78
	Проведение пути с минимальным числом пересечений .....	79
	Параллельная оптимизация пути по нескольким параметрам .....	80
7.3	Модификации волнового алгоритма .....	81
	Метод встречной волны .....	81
	Метод соединения комплексами: .....	81

Лучевой алгоритм трассировки .....	82
7.4 Эвристические алгоритмы трассировки .....	84
7.5 Особенности трассировки соединений в многослойных печатных платах .....	86
8 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ САПР. ПАКЕТ P-CAD 2000/04 .....	88
8.1 Общие сведения о системе проектирования P-CAD .....	88
Требования к оборудованию .....	90
8.2 Состав и структура системы P-CAD .....	90
Основные характеристики системы P-CAD. Общие характеристики:	95
Графический редактор схем P-CAD Schematic: .....	96
Графический редактор печатных плат P-CAD PCB: .....	96
Размерности величин .....	96
8.3 Создание библиотеки компонентов в САПР PCAD .....	97
Редактор символов Symbol Editor .....	98
Редактор корпусов Pattern Editor .....	100
Менеджер библиотек компонентов .....	103
Создание компонента с помощью Library Executive .....	108
Сохранение компонента в библиотеке .....	111
8.4 Создание модели схемы электрической принципиальной в редакторе Schematic P-CAD .....	112
Создание принципиальных схем .....	116
Проверка схем с помощью утилиты ERC .....	124
Вывод данных .....	125
8.5 Проектирование печатной платы в PCB P-CAD .....	126
Настройка конфигурации P-CAD PCB .....	126
Разработка печатных плат в PCAD PCB .....	128
Вывод данных .....	132
8.6 Автоматическая трассировка проводников .....	133
Программа автоматической трассировки Quick Route .....	133
Бессеточный трассировщик Shape-Based Router .....	137

ЧАСТЬ 2. ВЕСЕННИЙ СЕМЕСТР .....	143
1 ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ В САПР.....	143
1.1 Классификация задач конструкторского проектирования .....	143
1.2 Геометрический синтез.....	144
1.3 Геометрические модели.....	144
1.4 Комплексный подход к автоматизированному проектированию..	145
1.5 Автоматизированные САД/САМ/САЕ/PDM комплексы .....	146
2 СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ	
AUTOCAD 2000/2000I/2002/2004 .....	147
2.1 Назначение и возможности САПР AutoCAD .....	147
2.2 Пользовательский интерфейс .....	151
2.3 Способы ввода команд.....	158
2.4 Операции с файлами рисунков .....	159
2.5 Единицы измерения .....	162
2.6 Средства организации чертежа.....	166
2.7 Создание и оформление графических объектов для двухмерного моделирования.....	168
2.8 Редактирование графических объектов.....	179
ПОДГОТОВКА И ВЫПУСК ЧЕРТЕЖЕЙ. <i>ИДЕОЛОГИЯ СОЗДАНИЯ ЧЕРТЕЖЕЙ В AUTOCAD С ПРИМЕНЕНИЕМ ВИДОВЫХ ЭКРАНОВ.</i> СУЩЕСТВУЮТ ДВА ПРИНЦИПИАЛЬНО РАЗЛИЧНЫХ ПОДХОДА К ОФОРМЛЕНИЮ ЧЕРТЕЖЕЙ В AUTOCAD. ....	185
2.9 Трехмерное моделирование в системе AutoCAD .....	186
Моделирование поверхностей. Средства построения поверхностей	188
Средства редактирования поверхностей .....	188
Твердотельное моделирование. Средства построения твердотельных объектов .....	189
Средства редактирования твердотельных объектов.....	190
Общие средства редактирования трехмерных объектов.....	190
2.10 Пример трехмерного моделирования в AutoCAD .....	191
3 СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ T-FLEX CAD .....	195
3.1 Общие сведения о программном комплексе T-FLEX CAD.....	195
3.2 Особенности двумерного проектирования конструкций в T-FLEX	

CAD .....	200
Принципы параметризации чертежа.....	201
3.3 Трехмерное твердотельное проектирование .....	205
3.4 Создание трехмерной сборки электронного блока с использованием пакета T-FLEX CAD .....	209
4 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОГО ТЕХНОЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ.....	214
4.1 Комплексная автоматизация технологического проектирования	214
4.2 Методы автоматизации технологического проектирования .....	215
5 ТИПОВЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ ДИАЛОГОВОГО ПРОЕКТИРОВАНИЯ ТП .....	219
5.1 Функциональные математические модели .....	219
Оптимизация функциональных математических моделей.....	220
5.2 Структурно-логические модели ТП. Общие сведения.....	224
5.3 Табличная модель ТП .....	226
5.4 Сетевая модель ТП .....	229
5.5 Перестановочная модель ТП.....	230
5.6 Типовые решения в САПР технологических процессов.....	233
Типовые технологические процессы (по профессору А.А.Соколовскому).....	234
Групповые технологические процессы (по профессору С.П.Митрофанову).....	235
6 МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ .....	236
6.1 Метод прямого проектирования.....	236
6.2 Метод анализа .....	236
6.3 Метод синтеза в САПР технологических процессов.....	238
Реализация линейной стратегии проектирования в САПР технологических процессов.....	238
6.4 Оптимизация технологических процессов в САПР ТП .....	242
Структурная оптимизация ТП .....	243
Параметрическая оптимизация ТП .....	244
7 СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ МАРШРУТНОЙ ТЕХНОЛОГИИ ТЕХНОПРО .....	245

7.1 Общие сведения о системе ТехноПро .....	245
7.2 Методика автоматизированного проектирования ТП в САПР ТехноПро .....	247
7.3 Алгоритм проектирования ТП в САПР ТехноПро .....	249
Инструкция по созданию конкретного технологического процесса в САПР ТехноПро .....	260
8 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ САПР .....	265
8.1 Модели данных .....	265
Типы таблиц и ключей в реляционных базах данных .....	265
8.2 Нормализация данных .....	266
Первая нормальная форма .....	266
Вторая нормальная форма .....	267
Третья нормальная форма .....	267
8.3 Использование индексации таблиц .....	267
8.4 Этапы разработки базы данных .....	268
8.5 СУБД Access .....	268
Основные объекты Access .....	269
Таблицы .....	269
Типы данных .....	270
Запросы .....	270
Формы .....	270
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА .....	271
ОСНОВНАЯ .....	271
ДОПОЛНИТЕЛЬНАЯ .....	271
УЧЕБНО-МЕТОДИЧЕСКИЕ ПОСОБИЯ .....	272

## ЧАСТЬ 1. ОСЕННИЙ СЕМЕСТР

### 1 ВВОДНАЯ ЛЕКЦИЯ

#### 1.1 Необходимость применения САПР

Интегрирование РБ в мировую экономику невозможно без повышения эффективности производства за счет внедрения новых прогрессивных технологий и организации труда. Осуществить это возможно лишь при условии внедрения различных систем автоматизации, как разработки, так и изготовления и внедрения изделий.

Кроме того, постоянное ужесточение требований к электронной аппаратуре (ЭА) по точности, помехозащищенности, чувствительности, надежности ведет к значительному усложнению изделий и, в конечном счете, - к возникновению противоречия между требованиями к ЭА и конкретным производственным возможностям их удовлетворения:

1. чем сложнее ЭА, тем более трудоемка ее разработка (Сроки разработки увеличиваются);
2. увеличивается вероятность получения негативного результата (Быстрое моральное старение ЭА, при длительной разработке ЭА может устареть до окончания ее разработки).

Снижение сроков разработки достигается 2 путями:

##### **1. Увеличение количества разработчиков**

###### ***Недостатки:***

- а) Количество специалистов ограничено;
- б) уменьшается удельная производительность труда;
- в) снижается эффективность управления процессом проектирования и производства;
- г) Увеличивается стоимость разработки;
- д) Увеличивается число ошибок в проекте (добавляются ошибки согласования между отдельными частями проекта).

##### **2. Применение автоматизированного проектирования**

***Проектирование*** - процесс составления описания, необходимого для создания в заданных условиях еще не существующего объекта, на основе первичного описания этого объекта и (или) алгоритма его функционирования.

Модель удовлетворения общественных потребностей в изделиях (рисунок 1.1). На всех стадиях возможно применение автоматизированных систем.

Этапы «рождения», «жизни» и «смерти» изделия взаимосвязаны (см. рисунок 1.1) и решение задач по их оптимальному проектированию и производству

должно осуществляться комплексно на основе учета всех этапов. Необходимо целостное всестороннее рассмотрение всех вопросов проектирования и производства изделий с учетом их развития на других этапах в процессе взаимодействия с окружающей средой и человеческим обществом. Такой подход к проектированию и производству называется *системным*.

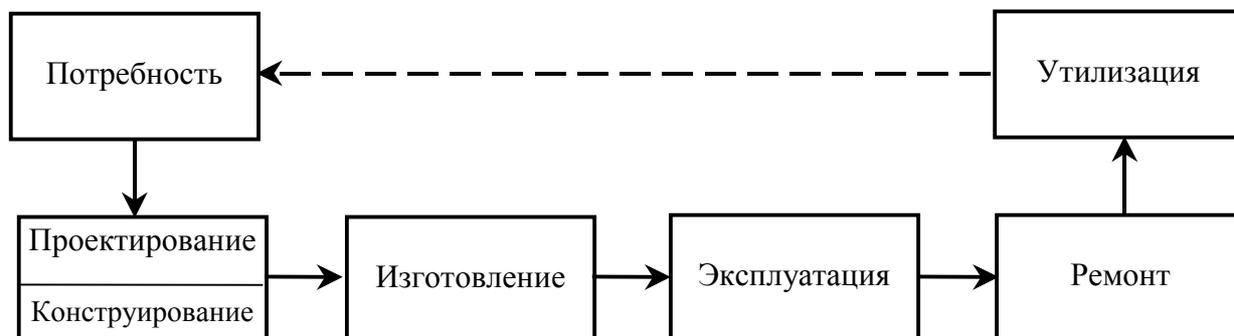


Рисунок 1.1 – Жизненный цикл изделий производства

Для избавления разработчиков от значительного объема «нетворческой» работы при проектировании ЭВМ используют *автоматизацию проектирования*, т.к. имеется возможность формального математического описания некоторых конструкторских задач и их унификация, а также уровень развития современных ЭВС позволяет создавать САПР.

При применении АПР происходит разделение сфер деятельности между человеком (инженером) и ЭВМ.

*Инженеру, как правило, остаются такие виды деятельности:*

- принятие решений и инженерное творчество;
- осмысление целей, проблем и содержания проекта;
- постановка задач проектирования;
- выбор критериев оценок при решении задач;
- выбор наилучшего решения и т. п.

*Основные задачи конструирования изделий ЭА, достаточно легко решаемые автоматизированным способом:*

1. оптимальное проектирование топологии и фотошаблонов ИМС, микросборок и печатных плат;
2. проектирование проводного монтажа;
3. компоновка (размещение) различных модулей низшего иерархического уровня в модули высшего иерархического уровня (например, ИМС на печатные узлы, ячейки в панели и т. д.);
4. выпуск графической и текстовой конструкторской документации.

### **Преимущества применения автоматизированного проектирования**

1. сокращение трудоемкости и сроков разработки конструкций ЭА,
2. повышение производительности труда инженеров в проектных организациях,
3. улучшение качества и технологического уровня изделий ЭА,

4. снижение стоимости разработки.
5. Уменьшение количества разработчиков
6. Использование уже готовых правильных решений (из баз данных)
7. Использование новейших достижений (облегчение обмена информации между базами + Internet ...).

**Система автоматизированного проектирования (САПР)** - комплекс средств автоматизации проектирования, взаимосвязанных с необходимыми подразделениями проектной организации или коллективом специалистов (пользователем системы), выполняющей автоматизированное проектирование.

**САПР (по ГОСТ 23.501.0-79)** – организационно-техническая система, состоящая из комплекса средств автоматизированного проектирования, связанных с подразделениями проектных организаций, и выполняющую автоматизированное проектирование.

**САПР** – инструмент проектирования, включающий различные обеспечения и предназначенный для автоматизированного проектирования на всех этапах от выдачи ТЗ до выпуска готового изделия.

Основная функция САПР - выполнение автоматизированного проектирования на всех или отдельных стадиях проектирования объектов и их составных частей.

В сфере производства ЭА можно выделить следующие системы автоматизации труда:

1. **Автоматизированные системы научных исследований**
2. **Системы автоматизированного проектирования**
3. **Автоматизированные системы ТПП**
4. Автоматизированные системы управления производством
5. Автоматизированные системы комплексных испытаний

### **Конструкторская и технологическая подготовка производства**

В процессе разработки сложных ЭО устройств различают следующие этапы:

1. **Системное проектирование,**
2. **Структурное проектирование,**
3. **функциональное,**
4. **схемотехническое,**
5. **конструкторское,**
6. **технологический синтез.**

При **системном** проектировании используются идеи и методы системного анализа. На основе многофакторного анализа проводится исследование ТЗ на разработку изделия и принимаются решения относительно методики построения и путей реализации процесса (**Используются системы автоматизации**

**научных исследований).**

При **структурном** проектировании на основании технического задания разрабатывается структурная схема, определяющая основные структурные части ЭА (устройства, блоки и т. п.), их назначение и взаимосвязи. Выбираются системы команд, диагностики и контроля, решаются вопросы обмена информацией между ЭА и внешними устройствами и абонентами процесса (**Используются системы автоматизации научных исследований**).

На данных этапах в основном используются творческие возможности человека (ЭВМ – как помощник в исследованиях).

При **функциональном** проектировании разрабатываются подробные функциональные схемы устройств проектируемой ЭВМ, которые разъясняют определенные процессы, протекающие в отдельных функциональных цепях или устройствах в целом и детализируют обмен информацией между ними.

В результате получают функциональные схемы устройств.

Критерии оптимизации:

- минимизация числа типов узлов
- достижение максимальной однородности блоков
- возможность эффективного моделирования
- простота диагностирования
- максимальный учет требований К и ТП.

Результат – получение математической либо физической модели устройства. Для ЭА – чаще математической.

Работа с моделями позволяет создать устройство требуемого качества с минимальными временными и материальными затратами (при помощи САПР).

При **схемотехническом** (логическом) проектировании разрабатываются подробные принципиальные схемы устройств, ориентированные на определенные системы элементов. Схемотехническое проектирование ЭА характеризуется большой трудоемкостью и, следовательно, требует большого количества разработчиков. Основные задачи схемотехнического проектирования хорошо формализуются и позволяют использовать машинные методы решения (**автоматизация проектирования**).

Задача:

- Покрытие функциональной схемы ячейками из заданного набора.

При **конструкторском** проектировании (или, иначе, конструировании) выбирается структура пространственных, энергетических и временных взаимосвязей частей конструкции, связей с окружающей средой и объектами, определяются материалы и виды обработки; устанавливаются количественные нормы (для связей, материалов и обработок), по которым можно изготовить изделие, соответствующее заданным требованиям (**часто используется САПР**).

Входная информация – функциональная и принципиальная схемы устройства.

На этом этапе необходимо решать следующие задачи:

1. Компоновка элементов схемы в ТЭЗы, в панели, в рамы ... (компоновка

высшего иерархического уровня элементами более низкого)

2. Размещение элементов в конструктивах по различным критериям

3. Распределение цепей по слоям платы. Трассировка соединений. Контроль правильности получения топологии.

4. Моделирование и расчет ЭМ и тепловых полей, механической прочности, надежности, вибропрочности ...

5. Автоматизированное получение конструкторской документации на изготовление ЭА.

При **технологическом синтезе** изготавливают отдельные устройства, блоки, происходит их наладка, общая сборка устройства и испытания (**САПР ТП**).

Цель:

- автоматизированное получение технологической документации
- разработка алгоритмов управления технологическим оборудованием
- разработка и изготовление технологической оснастки.

ГОСТ 2.103 — 68 устанавливает **стадии** разработки изделий и выпуска конструкторской документации при ОКР, а также определяет основные этапы выполнения работ на этих стадиях: 1).

- **техническое предложение,**
- **эскизный проект, (САПР)**
- **технический проект (САПР)**

относятся к проектным стадиям. Соответственно и документацию на изделие, выпускаемую на этих стадиях, называют *проектной*.

- **рабочий проект (САПР).**

**Техническое предложение** — совокупность конструкторских документов, которые содержат технические и технико-экономические обоснования целесообразности разработки документации изделия на основании анализа ТЗ и различных вариантов возможной реализации изделия, сравнительной оценки решений с учетом конструктивных и эксплуатационных особенностей разрабатываемого и существующих изделий, а также патентные исследования.

При разработке ЭА на стадии технического предложения прорабатываются следующие основные вопросы: обзор научно-технической литературы, патентов и нормативно-технических документов по рассматриваемой тематике; определение принципиальной возможности создания заказываемой ЭВМ; предварительные предложения по структуре ЭА и ее элементной базе; формулировка общих рекомендаций по разработке нескольких возможных вариантов конструкции ЭА; предварительное определение состава математического обеспечения; составление перечня организаций-соисполнителей ОКР, уточнение объемов, стоимости и сроков разработки, выработка предложений по уточнению ТЗ.

После уточнения, согласования и утверждения заказчиком ТЗ и приемки технического предложения приступают к разработке эскизного проекта.

**Эскизный проект** — совокупность конструкторских документов, содержащих принципиальные конструктивные решения, дающие общие представления об устройстве и принципе работы изделия, а также данные, определяющие

назначение, основные параметры и габариты изделия. На этой стадии выбранный вариант конструкции подвергается детальной проработке для выявления возможности наиболее полного удовлетворения всех предъявляемых к нему требований. Прорабатываются в основном следующие вопросы: теоретико-экспериментальные исследования по направлениям, намеченным на стадии технического предложения; выбор оптимального по ряду признаков варианта, подлежащего дальнейшей разработке, т. е. осуществляется выбор главного направления конструирования ЭА и ее составных частей; разработка технических решений, направленных на обеспечение показателей надежности ЭА, технологичности, стандартизации и унификации конструкций, эргономики, технической эстетики, техники безопасности и конкурентоспособности и др.; обоснование и выбор схемной реализации; составление технического задания на разработку новых компонентов, а также контрольно-измерительной аппаратуры; описание ЭВМ в целом и по устройствам; макетирование для отработки электрических схем, тепловых режимов, электромагнитной совместимости; предложения по уточнению технического задания.

После согласования, защиты и утверждения эскизный проект служит основанием для следующей проектной стадии — технического проекта.

**Технический проект** — совокупность конструкторских документов, содержащих окончательные технические решения, дающие полное представление об устройстве разрабатываемого изделия и исходные данные для разработки рабочей документации. Здесь проводится детальная отработка схемных и конструкторских решений (на уровне чертежей на все важные узлы, блоки, устройства), отрабатывается система математического обеспечения; разрабатываются технология изготовления составных частей ЭВМ и средства автоматизации ее проектирования и изготовления; выполняются пространственные компоновочные эскизы и макеты, позволяющие оценивать паразитные связи, тепловые режимы, удобство монтажа, ремонта, эксплуатации и защиту от внешних воздействий, изготавливаются узлы и блоки, которые проходят необходимые контрольные испытания; проверяются и оцениваются технологичность конструкций, степень соответствия их современному уровню микроэлектроники и комплексной микроминиатюризации, степень унификации и стандартизации и т. д.

После согласования, защиты и утверждения технического проекта переходят к рабочему проектированию ЭВМ.

**Рабочий проект** — это совокупность рабочей конструкторской и технологической документации, предназначенной для изготовления и испытания опытного образца, опытной партии, серийного (массового) производства. Рабочий проект является завершающей, наиболее ответственной стадией разработки ЭА.

На стадии рабочего проекта разрабатывается полный комплект КД, необходимых для изготовления, проверки, изучения и эксплуатации ЭА, технология изготовления отдельных узлов и ЭА в целом, изготавливаются опытные образцы ЭА, составляются программы предварительных (заводских) испытаний основных узлов и ЭА, проводятся эти испытания, корректируется по результатам испытаний документация.

При этом под *опытным образцом* понимают обычно ЭА, изготовленную по вновь разработанной рабочей документации для проверки его соответствия техническому заданию, проверки конструкторских решений, определения объема и характера последующей и необходимой корректировки конструкторских документов и подготовки технологического оснащения производства.

Для оценки качества опытного образца ЭА создается государственная или межведомственная комиссия. Она оценивает степень соответствия разработки всем требованиям технического задания, полноту и качество выполнения конструкторской и технологической документации, дает рекомендации по целесообразности передачи новых ЭА в серийное производство. При необходимости определяется перечень нужных доработок и проводятся дополнительные испытания ЭВМ. Приемкой опытного образца ОКР практически завершается, и комплекты конструкторской и технологической документации передаются на производство для организации серийного или массового выпуска ЭВМ.

В серийном или массовом производстве выполняется также ряд этапов конструкторских и технологических работ: изготовление и испытание опытной (установочной) серии изделий; корректировка конструкторских и технологических документов по результатам изготовления и испытания составных частей ЭВМ опытной серии; изготовление головной (контрольной) серии ЭВМ по полностью оснащеному технологическому процессу; корректировка конструкторских и технологических документов по результатам изготовления и испытания головной серии; непосредственно производство ЭА.

Для всех этапов проектирования изделия можно выделить типовые операции обработки информации:

1. Поиск и выбор из различных источников информации
2. Анализ
3. Расчет
4. Принятие проектных решений
5. Оформление проектных решений

Автоматизация 1-5 на всех стадиях проектирования – сущность современной САПР.

## **1.2 Классификация САПР (по ГОСТ 23501.8-80)**

### **По типу объекта проектирования.**

- 1) САПР изделий машиностроения;
- 2) САПР изделий приборостроения;
- 3) САПР технологических процессов в машино- и приборостроении;
- 4) САПР объектов строительства;
- 5) САПР технологических процессов в строительстве;
- 6) САПР программных изделий;

7) САПР организационных систем.

**По сложности объекта проектирования.**

- 1) простых объектов с числом составных частей до  $10^2$ ;
- 2) объектов средней сложности ( $10^2—10^3$ );
- 3) сложных объектов ( $10^3—10^4$ );
- 4) очень сложных объектов ( $10^4—10^6$ );
- 5) объектов очень высокой сложности (число составных частей свыше  $10^6$ ).

**По уровню автоматизации проектирования.**

- 1) низкоавтоматизированные (до 25% проектных процедур);
- 2) среднеавтоматизированные (25—50%);
- 3) высокоавтоматизированные (свыше 50%).

**По комплексности автоматизации проектирования.**

- 1) одноэтапные;
- 2) многоэтапные;
- 3) комплексные (все этапы).

**По характеру выпускаемых проектных документов.**

САПР, выпускающих документы:

- 1) на бумажной ленте и (или) листе;
- 2) на машинных носителях;
- 3) на фотоносителях (в виде микрофильмов, микрофиш, фотошаблонов и др.);
- 4) комбинированные (выполняют документы на двух носителях данных или более).

и т.д.

### **1.3 Составные части САПР (обеспечения) (ГОСТ 22487-77)**

1. *математическое,*
2. *программное,*
3. *лингвистическое,*
4. *информационное,*
5. *методическое,*
6. *организационное,*
7. **техническое.**

1-4 – программно-информационное

5,6 – организационно-методическое

**Математическое обеспечение САПР** — это совокупность математических методов, моделей и алгоритмов проектирования, необходимых для его выполнения.

В настоящее время для АПР ЭВМ широко применяются

– *модели и алгоритмы разбиения схем на подсхемы меньшей сложности,*

- алгоритмы размещения конструктивных модулей (перестановки, силовые и др.),
- алгоритмы разводки (трассировки) межсоединений различного уровня и т. д.

**Программное обеспечение САПР** – совокупность всех программ и эксплуатационной документации к ним, необходимых для выполнения АПР и представленных в заданной форме.

Различают

- 1) Общесистемное ПО (обеспечивает ввод, вывод и определение информации в процессе функционирования САПР),
- 2) Специальное (прикладное) ПО

В специальном ПО реализуются методы автоматизированного проектирования. Оно определяет специализацию системы, характер и степень использования ЭВМ в процессе конструирования, обеспечивают возможность модернизации и перестройки конкретной САПР.

Для подготовки и использования программных средств необходимо лингвистическое обеспечение.

**Лингвистическое обеспечение САПР** — это совокупность языков проектирования, включая термины и определения, правила формализации естественного языка и методы сжатия и развертывания текстов, необходимых для автоматизации проектирования.

Выделяют:

- а) Языки программирования
- б) Языки проектирования
- в) Языки управления

**Информационное обеспечение САПР** - совокупность сведений, необходимых для выполнения автоматизированного проектирования в заданной форме.

Т.е. основная функция ИО - обеспечение создания, поддержки и организации доступа к данным.

**Методическое обеспечение САПР** - комплект документов, устанавливающих состав и правила отбора и эксплуатации средств обеспечения проектирования.

Как правило, оно включает методические материалы по САПР, позволяющие проводить единую техническую политику по автоматизации проектирования на межотраслевом, отраслевом уровнях, уровне предприятия и т. д.

**Организационное обеспечение САПР** - комплект документов, устанавливающих состав проектной организации и ее подразделений, связи между ними, их функции, а также форму представления результата проектирования и порядок рассмотрения проектных документов.

**Техническое обеспечение САПР** включает универсальные и специализиро-

*ванные* технические средства электронно-вычислительной техники для автоматизации процессов проектирования, изготовления и контроля.

К *универсальным техническим средствам* относятся средства обработки информации, т.е. ЭВМ, а также средства ввода-вывода информации (принтеры, сканеры, плоттеры, графопостроители, манипуляторы, кодировщики и т.п.).

К *специализированным техническим средствам САПР* относят:

- автоматизированные рабочие места (АРМ) (за рубежом используется термин «рабочая станция»),
- пункты выпуска документации,
- автоматизированные средства для изготовления и контроля конструкций ЭВМ (современные САПР позволяют кроме КД изготавливать макеты проектируемых устройств, ПО для ГАП и т.п.).

## **1.4 Подсистемы САПР**

Специализация части САПР на обслуживании проектных задач одного этапа проектирования приводит к выделению этой части как п/системы САПР.

Специализация САПР затрагивает, в основном, МО, ПО, ЛО, ТО.

Для САПР ЭА выделяют подсистемы (указывалось ранее):

1. Системотехнического проектирования
2. Схемотехнического проектирования
3. Конструкторского проектирования
4. Технологического проектирования

Для каждой п/системы есть свои пакеты прикладных программ. В настоящее время стремятся разрабатывать комбинированные ППП, включающие большинство п/систем.

П/системы САПР можно разделить на 2 группы:

1. Проектирующие
  2. Обслуживающие
- 1) – обеспечивают выполнение проектных работ
  - 2) – предназначены для поддержки работоспособности проектирующих систем.

В зависимости от отношения к объекту проектирования САПР делятся на:

1. объектные (выполняют одну или несколько процедур непосредственно зависящих от объекта проектирования)
2. инвариантные (унифицированные объектные процедуры и операции)

## **1.5 Основные требования и принципы создания САПР**

Необходимым и достаточным условием создания САПР на предприятии является количественный или качественный выигрыш от ее применения с уче-

том дополнительных затрат, которые она вызывает.

*САПР должна иметь возможности (требования):*

1. последовательного расширения и совершенствования системы
2. активной связи специалист – система
3. оперирования оптимальными взаимозаменяемыми алгоритмами
4. специализацию системы на проектирование (ЭА, РЭА, ИМС, ...)
5. увеличения мощности системы
6. легкой настраиваемости
7. стыковки со специальными устройствами
8. изменения критериев оптимизации
9. расширения и дополнения библиотек программ
10. свободного доступа к данным на всех этапах проектирования
11. изготовления КД и ТД

*Работоспособность САПР зависит от:*

1. Уровня унификации входной и выходной информации
2. Единства методов записи информации на носителях
3. Правильности распределения памяти ЭВМ
4. Правильности выбора алгоритмического языка

Для удовлетворения требований к САПР при ее создании должны соблюдаться следующие основные **принципы**:

1. Принцип системности.

Заключается в том, что при создании САПР взаимные связи между ее п/системами должны обеспечивать работоспособность и целостность САПР как единой системы.

2. Принцип включения.

Его реализация позволяет включать данную САПР в систему более высокого уровня.

3. Принцип развития (открытости).

Необходимо предусматривать возможность совершенствования, развития и дополнения, а также обновления основных компонентов м п/систем.

4. Принцип комплексности.

Позволяет обеспечить взаимосвязанное проектирование, как отдельных элементов, так и всего объекта в целом. На всех стадиях осуществить согласование, увязку и контроль характеристик проектируемых элементов м системы в целом.

5. Принцип модульности.

Относится к ПО и предполагает построение ПО САПР в виде отдельных программных модулей.

Это обеспечивает возможность включения или исключения отдельных программ, реализующих те или иные процедуры, без нарушения функционирования всей системы САПР.

6. Принцип информационного единства.

Отделение всех видов данных от программ и использование в п/системах

САПР единых условных обозначений, терминов и способов представления информации в соответствии с действующей нормативной документацией.

Реализация данного принципа предполагает создание и функционирование самостоятельной БД.

7. принцип совместимости.

Использование языков, кодов, символов, информационных и технических характеристик связей между п/системами, которые должны обеспечить их совместное функционирование, а также возможность развития всей системы САПР в целом.

8. Принцип стандартизации.

Осуществление унификации, типизации и стандартизации п/систем и компонентов, инвариантных к проектируемым объектам и проектным процедурам.

## 2 Математическое обеспечение САПР

**Математическое обеспечение (МО)** – это совокупность математических методов и моделей алгоритмов проектирования, необходимых для их выполнения.

Основу МО САПР составляет математический аппарат для моделирования, анализа и оптимизации.

В МО САПР выделяют:

1. Специальную часть
2. Инвариантную часть

1) – отображает специфику объекта проектирования, физические и информационные особенности его функционирования. Специальная часть тесно привязана к этапам проектирования.

2) – включает методы и алгоритмы, слабо связанные с особенностями математических моделей и используется на различных этапах проектирования.

Основу инвариантной части в 1-ю очередь составляют методы параметрического синтеза и многовариационного анализа.

### 2.1 Требования к МО САПР

1. Универсальность
2. Алгоритмическая надежность
3. Точность
4. Затраты машинного времени (min)
5. Используемая память

1) Под **универсальностью** МО понимается его применимость к широкому классу проектируемых объектов.

Высокая степень универсальности МО необходима для того, чтобы САПР была применима к большинству объектов, проектируемых на предприятии. Степень универсальности не имеет количественной оценки.

Если методы и модели сформулированы интуитивно, то они называются **эвристическими**.

2) Свойство компонента МО давать при его применении правильные результаты называется **алгоритмической надежностью**.

Количественной оценкой алгоритмической надежности служит вероятность получения правильных результатов при соблюдении оговоренных ограничений на применение метода.

3) **Точность** определяется по степени совпадения расчетных и истинных результатов.

Оценка точности производится с помощью специальных вычислительных экспериментов, в которых создаются условия отдельной оценки погрешностей, вносимых математическими моделями элементов, алгоритмами анализа и оптимизации. В этих экспериментах используются специальные задачи (тестовые).

Количественная оценка погрешности результата решения тестовой задачи есть одна из нормалей вектора относительных погрешностей.

4) Универсальные модели и методы характеризуются большим объемом вычислений, растущих с ростом размерности решаемых задач.

**Затраты машинного времени** являются главным ограничивающим фактором при повышении сложности проектируемых на ЭВМ объектов.

Требования высокой степени универсальности, алгоритмической надежности, точности с одной стороны и малых затрат машинного времени с другой – основное противоречие при создании МО САПР. Поэтому при создании САПР целесообразно использовать библиотеки с наборами моделей и методов, перекрывающих потребности всех пользователей САПР.

5) Затраты **используемой памяти** являются вторым фактором после затрат машинного времени экономическим показателем САПР. Она определяется объемом программы и используемых массивов.

Несмотря на значительный объем памяти в современных ЭВМ требования по экономичным затратам памяти остаются актуальными, т.к. современные системы работают в мультипрограммном режиме и задачи с большим запросом объема памяти получают более низкий приоритет.

## 2.2 Методы повышения эффективности МО САПР

Увеличение эффективности САПР может быть достигнуто следующими путями:

1. разработкой экономичных моделей и алгоритмов, имеющих частный характер
2. совершенствования используемых *общих принципов*, создания МО эффективного по затратам машинного времени и памяти.

К таким принципам относятся:

- 1) учет разряженности матриц
- 2) исследование сложных систем по частям
- 3) макро моделирование
- 4) событийность анализа
- 5) рациональное использование эвристических способностей человека в интерактивных (диалог) процедурах

- 1) – позволяет во многих алгоритмах в которых используются операции

над матрицами добиться существенного сокращения затрат времени и памяти. Т.к. в памяти ЭВМ хранят только ненулевые элементы матриц и выполняют арифметические действия только над ними.

Сильно разреженными матрицами при проектировании ЭА являются:

а) матрицы коэффициентов систем разностных уравнений при анализе п/проводниковых компонентов электронных схем

б) матрицы Якоби – в мат. Моделях электронных схем

в) матрицы смежности и инцидентности в задачах компоновки, трассировки и размещения.

Если учитывать разреженность матриц, то затраты времени и памяти можно сделать линейно зависимыми от показателя сложности анализируемого объекта (если нет – то квадратичная либо кубическая зависимость).

2) – реализуется в диаконтических методах исследования.

Диаконтические методы исследования отличаются от блочно-иерархических использованием структурных особенностей исследуемых схем и выражающих их матриц. Принятие упрощений производится расчленением математических моделей на части, которые исследуются самостоятельно.

3) – Макромоделирование связано с рациональным выбором мат. модели. Оно реализует возможность системы использования при анализе одного и того же объекта нескольких моделей, различающихся сложностью, точностью и полнотой отображения объекта.

Экономическая эффективность системы повышается путем деления схемы на более крупные блоки с использованием для них макромоделей.

4) – событийность анализа заключается в том, что при иммитации процессов, протекающих в исследуемом объекте в каждый отдельный момент времени вычисления производятся только для небольшой части мат. модели объекта. Эта часть включает в себя те элементы, состояние которых на данном временном шаге может измениться.

Использование данного принципа повышает экономичность анализа на функциональном, системном и логическом этапах проектирования.

5) – позволяет инженеру (опытному) вмешиваться в ход вычислений и выбирать наиболее перспективные продолжения на основе эвристических оценок. Процедура наиболее эффективна на этапе конструкторского проектирования.

### 2.3 Основы теории графов

Под **графом**  $G(X, U)$  понимают совокупность непустого множества  $X$  и изолированное от него подмножество  $U$ , возможно нулевое, представляющее собой множество всех упорядоченных пар  $x_i x_j$ , где  $x_i, x_j$  принадлежат  $X$ ;  $i, j = 1 \dots n$ , где  $n$  – **мощность** множества.

Элементы множества  $X$  и  $U$  соответственно называются **вершинами** и

**ребрами** графа.

Существует 3 вида графов:

1. Неориентированные
2. Ориентированные
3. Смешанные

Граф  $G(X, U)$  называется **неориентированным**, если для каждого его ребра несущественен порядок двух его концевых вершин.

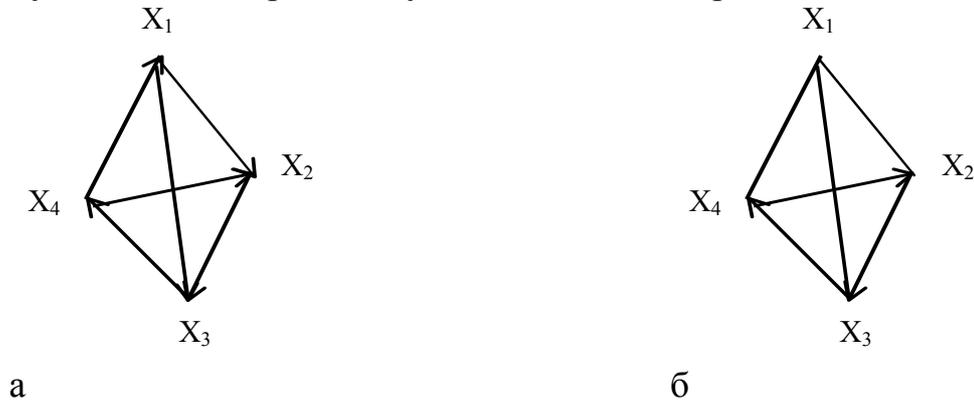


Рисунок 2.1 – Примеры графов:  
а – ориентированный, б - смешанный

Граф, у которого 2 вершины соединены более чем одним ребром – **мультиграф**.

Ребра, у которых обе концевые вершины совпадают, называются **петлями**.

Вершина неинцидентная (не принадлежит) никакому ребру называется **изолированной**.

Число ребер инцидентных некоторой вершине  $x_i$  называется **степенью вершины**.

Граф, состоящий только из изолированных вершин называется **нульграфом**.

Граф **конечен**, если содержит конечное число вершин и ребер.

Конечный граф, у которого отсутствуют петли и изолированные вершины называется **регулярным**.

Граф называется **однородным степени  $t$** , если степень всех его вершин  $= t$ .

Пример однородного графа ( $t=4$ )

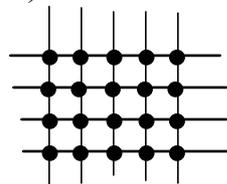


Рисунок 2.2

Данный бесконечный граф широко используется для трассировки печатных плат. Он позволяет разбивать коммутационное поле платы на элементарные ячейки.

Граф, все вершины которого попарно смежны называется **полным графом**.

Полный граф, у которого при каждой вершине имеется петля, называется **плотным**.

Граф, в котором перемещаясь по ребрам из вершины в вершину можно попасть в каждую вершину называется **связным** графом.

Число, характеризующее разность между числом верши графа (мощностью)  $n$  и числом компонент связности  $p$  называют **рангом графа** ( $R(G)$ ).

Один и тот же граф может иметь различные геометрические реализации (получаются **изоморфные** графы).

**Циклом** называется последовательность ребер, при которой в результате обхода вершин графа по этим ребрам возвращаются в исходную вершину.

Последовательность ребер при переходе от одной вершины к другой называется **цепью**.

Цикл – это замкнутая цепь.

**Эйлеров цикл** – это цикл, в котором содержатся все ребра графа.

Задача отыскания эйлерова цикла имеет важное прикладное значение при выборе рационального пути движения рабочего инструмента, позволяя осуществить обход контура сложной геометрической фигуры без отведения рабочего инструмента, т.е. без холостых ходов.

Цикл называют **гамильтоновым**, если он проходит через каждую вершину графа только один раз.

Связной неориентированный граф, не содержащий циклов, называется **деревом**.

Несвязной граф без циклов, отдельные компоненты связности которого являются деревьями называется **лесом**.

Под **расстоянием** между вершинами графа понимается длина кратчайшей цепи. Соединяющей эти вершины.

**Диаметр** графа – это максимальное расстояние между вершинами графа.

Объект  $H(X, E)$  считается **гиперграфом**, если он состоит из множества вершин  $X$  и множества ребер  $E$ , причем каждое ребро  $e_i$ , принадлежащее  $E$  является некоторым подмножеством множества  $X$ . Т.е. множество  $X$  должно включать любое ребро  $e_i$ .

При этом каждое ребро может соединять не только две вершины, но и любое подмножество множества вершин графа.

## 2.4 Матричное представление графов

Удобной формой описания графов является представление их в виде **матриц**. Методики формальной обработки матриц хорошо разработаны.

### 1. Матрица смежности.

Если задан граф  $G(X, U)$ , то ему можно поставить в соответствие квадратную матрицу (матрицу смежности) размерностью  $n \times n$ , где  $n$  – мощность мно-

жества вершин графа

$$A = \|a_{ij}\|_{n \times n}, \text{ где}$$

$$a_{ij} = \begin{cases} m(x_i, x_j), & \text{если } x_i, x_j \text{ — смежные} \\ 0, & \text{если — нет} \end{cases}$$

где  $m$  — кратность смежных ребер.

Пример:

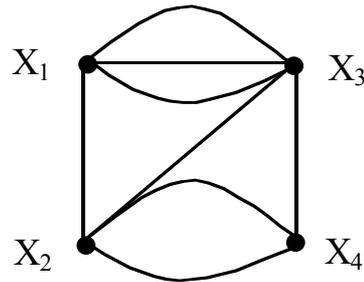


Рисунок 2.3

$$A = \begin{vmatrix} & x1 & x2 & x3 & x4 \\ x1 & 0 & 1 & 3 & 0 \\ x2 & 1 & 0 & 1 & 2 \\ x3 & 3 & 1 & 0 & 1 \\ x4 & 0 & 2 & 1 & 0 \end{vmatrix}$$

## 2. Матрица весовых соотношений.

При решении целого класса задач проектирования ЭА приходится оперировать матрицами, которые строятся аналогично матрицам смежности, но значения их элементов определяются весом ребра графа. Такие матрицы называются **матрицами весовых соотношений**.

$$C = \|c_{ij}\|_{n \times n}, \text{ где}$$

$$c_{ij} = \begin{cases} T_{ij}, & \text{если } x_i, x_j \text{ — смежные} \\ 0, & \text{если — нет} \end{cases}$$

где  $T_{ij}$  — вес связи.

## 3. Матрица длин.

Это квадратная матрица

$$D = \|d_{ij}\|_{n \times n}, \text{ где}$$

$$d_{ij} = \begin{cases} L_{ij}, & \text{если } x_i, x_j - \text{смежные} \\ 0, & \text{если - нет} \end{cases}$$

где  $L_{ij}$  – длина ребра.

Для примера однородного графа:

$$L_{ij} = \sqrt{(s_i - s_j)^2 + (t_i - t_j)^2},$$

где  $(s_i, t_i)$  – координата 1-й точки

$(s_j, t_j)$  – координата 2-й точки

При расчете конструктивных параметров ЭА, в котором производится вычисление произвольной координаты:

$$L_{ij} = (s_i - s_j)^k + (t_i - t_j)^k.$$

Матрицы **1-3** используются в конструировании ЭА,

Матрица **3** используется в задачах трассировки.

#### 4. Матрица инцидентности.

Представляет собой прямоугольную матрицу

$$B = \|b_{ij}\|_{n \times r}$$

размерностью  $n \times r$ , где  $n = |x|$ ,  $r = |u|$ .

Строки матрицы соответствуют вершинам, а столбцы – ребрам графа.

$$b_{ij} = \begin{cases} 1, & \text{если } x_i, u_j - \text{смежные} \\ 0, & \text{если - нет} \end{cases}$$

Пример:

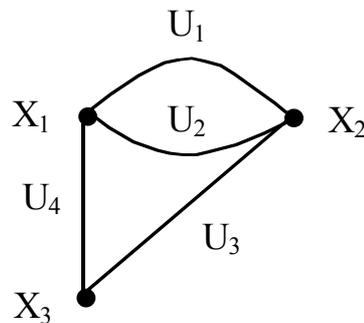


Рисунок 2.4

$$B = \begin{vmatrix} & u1 & u2 & u3 & u4 \\ x1 & 1 & 1 & 0 & 1 \\ x2 & 1 & 1 & 1 & 0 \\ x3 & 0 & 0 & 1 & 1 \end{vmatrix}$$

### 5. Матрица смежности ребер.

Эта матрица, элементы которой образуются по правилу

$$W = \|w_{ij}\|_{n \times n}, \text{ где}$$

$$w_{ij} = \begin{cases} 1, & \text{если } u_i, u_j \text{ – смежные} \\ 0, & \text{если – нет} \end{cases}$$

Пример:

Граф – из предыдущего примера

$$W = \begin{vmatrix} & u1 & u2 & u3 & u4 \\ u1 & 0 & 1 & 1 & 1 \\ u2 & 1 & 0 & 1 & 1 \\ u3 & 1 & 1 & 0 & 1 \\ u4 & 1 & 1 & 1 & 0 \end{vmatrix}$$

## 2.5 Основы теории алгоритмов

**Алгоритм** – это конечная совокупность точно заданных правил решения произвольного класса задач.

Алгоритм состоит из отдельных конечных действий – **шагов**.

Существует три основных универсальных алгоритмических моделей:

1. Понятие алгоритма связывается с вычислениями и числовыми функциями.
2. Алгоритм представляется как некоторое детерминированное устройство, способное выполнять в определенные моменты времени типовые (простейшие) операции.
3. Производится преобразование слов произвольных алфавитов.

**Детерминированный** алгоритм - если он выражен системой правил, одно-

значно определяющих результат процесса при заданных исходных данных.

Если правила неоднозначны и результаты можно представить только статистически, то такой алгоритм называется **вероятностным** или **стахостическим**.

Если правило нельзя задать ни вероятностно, ни детерминированно, но можно сформировать содержательные указания о целенаправленности процесса, то такой алгоритм называется **эвристическим**.

Расшифровка укрупненных операторов алгоритма в командах языка ЭВМ называется **программированием**, а запись алгоритма на входном языке ЭВМ - **программой**.

Эвристическое программирование не гарантирует нахождение оптимального варианта, а только – квазиоптимальное решение.

### **Свойства алгоритмов**

1. **Массовость** – свойство алгоритма отображать широкий класс процессов.

2. **Результативность** - свойство алгоритма обеспечивать получение результата через конечное число шагов.

3. **Область применения** – множество процессов, для которых алгоритм результативен.

4. **Определимость** - свойство алгоритма, заключающееся в том, что каждый его шаг определяется точно.

5. **Алгоритм** имеет вход и выход.

6. **Алгоритмы являются эквивалентными** – если совпадают их области применимости и результаты обработки любого процесса из данной области.

7. **Алгоритмы равны** – если равны соответствующие им операторы и совпадают системы правил, задающие действия этих алгоритмов.

### **Эффективность алгоритмов:**

Оценка максимального числа элементарных операций, выполняемых при работе алгоритма – эффективность.

Оценка эффективности зависит от длины исходных данных (числа элементов схемы, цепей ...) и характеризуется числом операций  $N$  и общим объемом памяти  $V$ , необходимых для его реализации.

Общее время реализации алгоритма:

$$T = \sum_{i=1}^n Q_i t_i, \quad (2.1)$$

где  $Q_i$  – число операций  $i$ -го типа,  $n$  – число типов операций в алгоритме,  $t_i$  –

время выполнения  $i$ -й операции.

Если ввести условную простейшую элементарную операцию, то можно определить общее число приведенных к ней операций:

$$N_{\text{э}} = \sum_{i=1}^n B_i Q_i, \quad (2.2)$$

где

$$B_i = \frac{t_i}{t_{\text{э}}},$$

$t_{\text{э}}$  – время выполнения элементарной операции.

Тогда общее время реализации алгоритма:

$$T = N_{\text{э}} \cdot t_{\text{э}}. \quad (2.3)$$

Объем памяти, кот. необходимо зарезервировать в ЭВМ для реализации алгоритма определяется, как

$$V = V_{\text{П}} + V_{\text{ВХ}} + V_{\text{ПП}} + V_{\text{ВЫХ}}, \quad (4.4)$$

где  $V_{\text{П}}$  – объем памяти для размещения программы,  $V_{\text{ВХ}}$ ,  $V_{\text{ВЫХ}}$  - объем памяти для размещения исходной входной и выходной информации,  $V_{\text{ПП}}$  - объем памяти, необходимой для размещения промежуточной информации.

Коэффициент сложности алгоритма тем выше, чем выше  $N_{\text{э}}$  и  $V$ .

## 2.6 Способы записи алгоритмов

Практически все алгоритмы состоят из логических и арифметических операторов.

Арифметические – выполняют непосредственное преобразование информации.

Логические - определяют последовательность выполнения арифметических операторов.

### 1) Операторный алгоритм Ван-Хао

Алгоритм задается последовательностью приказов специального вида. Каждый приказ имеет определенный номер и содержит следующие указания: какую операцию следует выполнить над заданным объектом и приказ с каким номером следует далее выполнить действие над результатом данной операции.

Общий вид приказа:

$$i \left\{ \begin{array}{|c|c|c|} \hline \omega & \alpha & \beta \\ \hline \end{array} \right.$$

где  $i$  – номер приказа,  $\omega$  - элементарная операция над объектом,  $\alpha$  и  $\beta$  - но-

мера некоторых приказов.

Выполнить приказ  $i$  над числом  $X$  в операторе алгоритма – значит найти число  $W(X)$  и затем перейти к выполнению приказа  $\alpha$ . Если значение  $W(X)$  не определено, то перейти к выполнению над числом  $X$  приказа с номером  $\beta$ .

## 2) форма логической схемы алгоритма.

логической схемой алгоритма называются выражения, составленные из операторов и логических условий, следующих один за другим. После каждого логического условия ставится стрелка, которая оканчивается у какого-либо оператора ( $\uparrow$  - начало,  $\downarrow$  - конец).

Для записи алгоритмов используют основные типы операторов:

- арифметические операторы (обозначаются начальными заглавными буквами латинского алфавита),
- операторы проверки логических условий (малые буквы латинского алфавита),
- операторы переадресации (обозначается буквой F (..). В скобках указан изменяемый адрес или параметр),
- операторы переноса,
- операторы формирования.

Например

$$A_0 p_1 \uparrow^1 A_1 \downarrow^1 p_2 \uparrow^2 A_2 A_3 \downarrow^2 A_4 A_k$$

где –  $A_0$ ,  $A_k$  – операторы начала и конца,

$A_1 \dots A_4$  – операторы,

$p_1$ ,  $p_2$  – логические условия.

Если  $p_1 = 1$ , то произойдет переход на оператор  $A_1$ , если  $= 0$ , то произойдет переход на оператор  $p_2$ .

Аналогично – для  $p_2$ .

## 3) В виде структурной схемы алгоритма.

(обычно – в виде различных блоков, связанных логическими связями между собой).

## 4) Словесное задание алгоритмов

3) Алгоритм расчленяется на отдельные блоки, которые отображаются в виде геометрических фигур. Блоки нумеруются и внутри них указываются действия, которые записываются в виде формул или словестно.

Блоки соединяются стрелками, показывающими связи между ними.

Если логические условия передают управления другим блокам, то на стрелках этих блоков указываются условия, при которых процесс разветвляется.

Кроме этого имеются блоки ввода и вывода.

Примеры записи вершин графа при графической записи алгоритма:

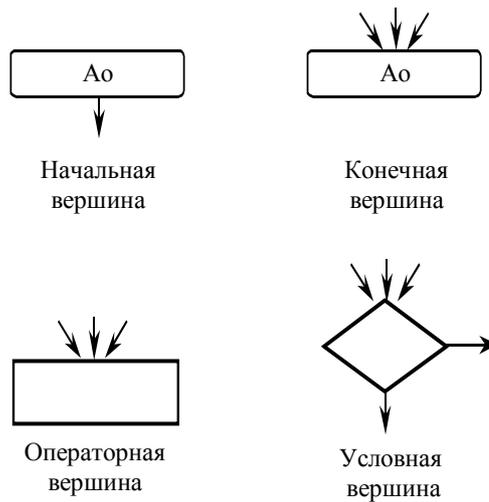


Рисунок 2.3

Граф-схема ранее записанного алгоритма (см. рисунок 2.4).

**Достоинства:**

1. Обеспечивается возможность обмена структурными схемами алгоритмов между специалистами.
2. Обеспечивается наглядное чтение и понимание алгоритмов.
3. Уменьшается число ошибок при программировании.

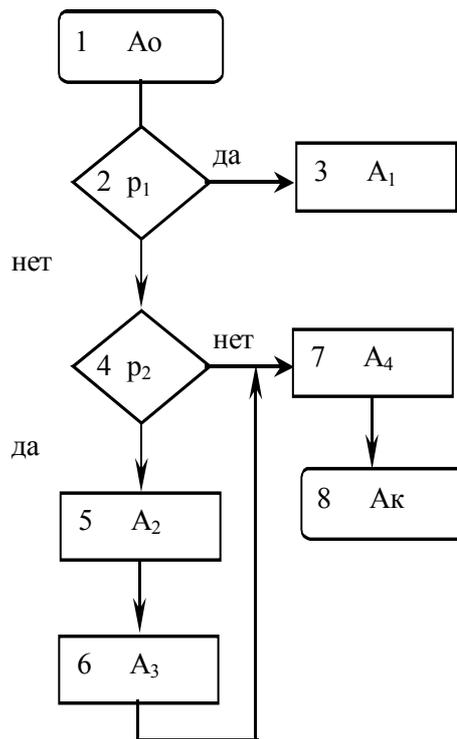


Рисунок 2.4

**4) Словесное задание алгоритмов**

При данном способе задания алгоритма перечисляются блоки алгоритма и в них записываются логические условия.

### 3 Формальное описание коммутационных схем

Любая коммутационная схема ЭА состоит из набора элементов заданным образом соединенных между собой. Поэтому схему можно представить как некоторое подмножество элементов  $X_L$ :

$$X = \{x_1, x_2 \dots x_n\}$$

Соединенных между собой цепями из множества  $E$

$$E = \{e_1, e_2 \dots e_m\}$$

Представляя гиперграф  $H(X, E)$  матрицей инцидентности получаем удобную форму представления схемы в памяти ЭВМ.

Электрическую схему задают также в виде **матрицы цепей**:

$$T = \left\| t_{ij} \right\|_{n \times m}$$

Строки матрицы  $T$  соответствуют элементам схемы, а столбцы – контактам элементов.

Если в схеме имеются элементы с различным числом контактов (наиболее часто встречается), то в качестве  $m$  принимается число контактов элементов с максимальным числом контактов.

**Пример:**

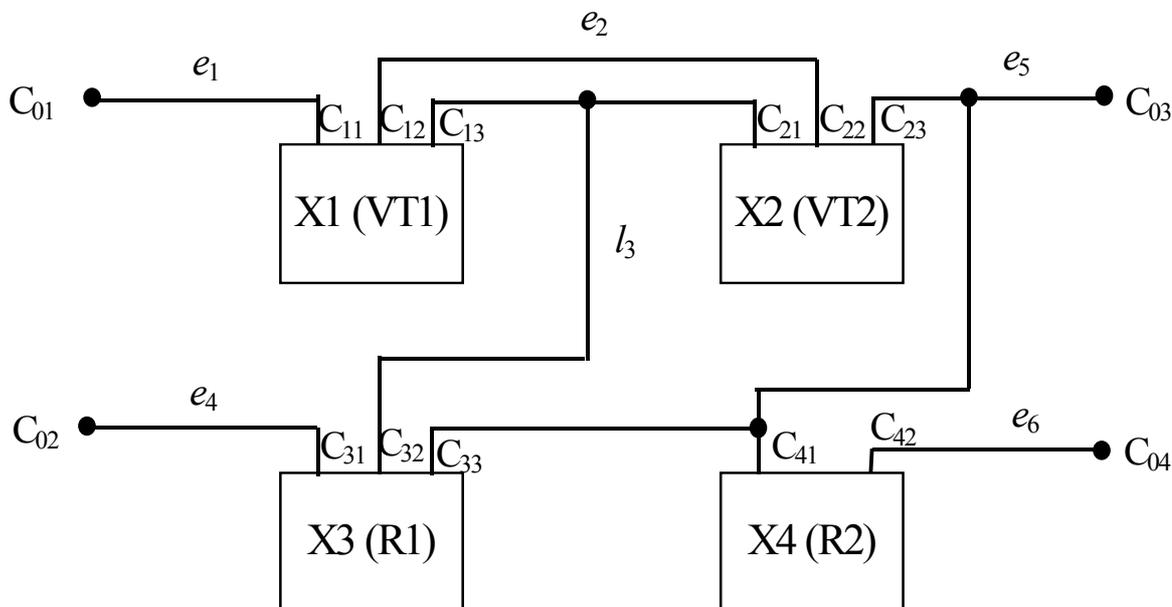


Рисунок 3.1 Схема электрическая

$$B = \begin{matrix} & l1 & l2 & l3 & l4 & l5 & l6 \\ X0 & 1 & 0 & 0 & 1 & 1 & 1 \\ X1 & 1 & 1 & 1 & 0 & 0 & 0 \\ X2 & 0 & 1 & 1 & 0 & 1 & 0 \\ X3 & 0 & 0 & 1 & 1 & 1 & 0 \\ X4 & 0 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

$$T = \begin{matrix} & 1 & 2 & 3 & 4 \\ X0 & l1 & l4 & l5 & l6 \\ X1 & l1 & l2 & l3 & 0 \\ X2 & l3 & l2 & l5 & 0 \\ X3 & l4 & l3 & l5 & 0 \\ X4 & l5 & l6 & 0 & 0 \end{matrix}$$

### 3.1 Основные модели представления коммутационной схемы ЭА в памяти ЭВМ

#### Модель 1

Схема строится в виде мультиграфа  $G(X,U)$ , где  $X$  – множество элементов схемы,  $U$  – множество ребер, каждое из которых соответствует соединению между  $i$ -тым и  $j$ -тым элементами схемы.

Обычно модель представляется в ЭВМ в виде матрицы смежности.

#### *Характеристики модели:*

1. Модель широко используется при решении задач компоновки и размещения.
2. Описание данной модели обладает избыточностью, т.к. число ребер, инцидентных отдельным вершинам модели оказывается больше фактического осуществления трассировки.
3. Алгоритм обработки достаточно прост.
4. Переход от объекта к модели достаточно прост.
5. *Недостаток:* Требуется значительный объем памяти, пропорциональный квадрату числа элементов схемы.

#### Модель 2.

Вершины графа  $G(X,U)$  сопоставляются выводам элементов, а соединения между ними – ребрам.

Модель применяется для решения задач трассировки.

Ее характеристики аналогичны характеристикам **Модели 1.**

### **Модель 3.**

Электрическая схема представляется гипер-графом, который в памяти ЭВМ представляется матрицей инцидентности.

Характеристики модели:

1. Обычно используется при компоновке и размещении элементов.
2. Позволяет достаточно точно определить основные конструктивные параметры устройства.
3. Установление однозначности объекта и модели не вызывает затруднений.
4. *Недостатки:* Алгоритмы обработки модели достаточно сложны.
5. Требуется большой объем памяти ЭВМ.

### **Модель 4.**

Для построения модели необходимо пронумеровать все цепи схемы. Вершины модельного графа сопоставляются элементам схемы. Выводы контактов элементов и цепи нумеруются.

Исходными данными для ввода является матрица цепей.

Характеристики модели:

1. Универсальна.
2. Обладает достаточной степенью детализации и позволяет точно описать объект.
3. Объем памяти значительно меньше, чем у модели 1-3.
4. *Недостатки:* Алгоритмы обработки достаточно сложны (сложнее чем у моделей 1 и 2).
5. Переход от объекта к модели достаточно сложен.

## **3.2 Основная модель монтажного пространства**

**Монтажным пространством** элементов конструкций называется некоторая область, ограниченная габаритами этих элементов.

В монтажном пространстве данного модуля размещаются элементы низшего иерархического уровня и осуществляется их электрическое соединение.

Двумерное монтажное пространство называется **монтажным полем**.

Различают **регулярное** и **нерегулярное** монтажное поле (МП).

**Регулярное** МП имеет прямоугольную форму и равные по размеру элементы, расположенные с равным шагом по горизонтали и вертикали.

**Нерегулярное** МП характеризуется тем, что элементы имеют разные размеры и не имеют точно определенных посадочных мест.

Минимальный размер ячейки определяется выражением:

$$\delta \geq h + 2s ,$$

где  $h$  – ширина проводника,  $s$  – минимальное расстояние между проводниками.  
Общее число дискретных ячеек:

$$N = n \cdot m$$

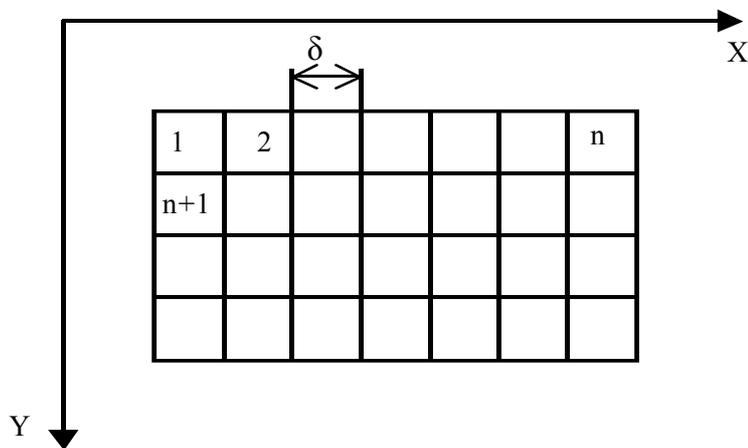


Рисунок 3.2 – Модель монтажного пространства

Место любого  $i$ -го дискрета на монтажном поле однозначно может быть указано его координатами  $(x_i, y_i)$  в системе дискретных координат, либо индексом  $I$ , значение которого через координаты  $x_i$  и  $y_i$  выражается следующим образом:

$$i = x_i + (y_i - 1) \cdot n$$

В зависимости от того, проходит ли через текущий дискрет проводник, либо в каком направлении проходит этот проводник можно рассматривать некоторое конечное множество состояний каждого дискрета.

Каждому дискрету можно поставить в соответствие некоторое число, код.

Машинным эквивалентом дискретного монтажного поля может служить некоторый двумерный массив  $B(X, Y)$ , значения каждого элемента которого соответствуют состоянию дискрета с координатами  $X, Y$ .

Уменьшая шаг координатной сетки  $\delta$  можно с какой угодно точностью описать монтажную плату.

Однако при значительно уменьшающемся размере дискрета увеличивается объем памяти.

Аналогично можно поставить в соответствие каждой ячейке вершину графа,

тогда модель можно описать графом  $G(X, U)$ , вершины которого соответствуют вершинам дискретов, а ребра – отображают связи между дискретами.

Модель монтажного пространства описывается также матрицей расстоя-

ний:

$$D = \left\| d_{ij} \right\|_{n \times m},$$

где

$$d_{ij} = \begin{cases} L_{ij}, & \text{если } x_i, x_j - \text{смежные} \\ 0, & \text{если - нет} \end{cases}$$

где  $L_{ij}$  – длина ребра.

### 3.3 Модель электрической схемы

Правила построения модели электрической схемы включают те же задачи, что и при построении модели монтажного пространства:

Производится формализация задачи, т.е. осуществляется переход от элементов схемы к графу, а граф представляется одним из матричных представлений.

Схему рассматривают как некоторое множество элементов  $X = \{x_1, x_2, \dots, x_n\}$ , соединенные между собой цепями из множества  $E = \{e_1, e_2, \dots, e_m\}$ .

Каждый элемент схемы имеет некоторое множество соединительных выводов, которые называются множеством контактов  $C = \{c_1, c_2, \dots, c_s\}$ .

Кроме контактов элементов в схеме имеются внешние контакты  $C_0$ , которые осуществляют связь рассматриваемой схемы с другими элементами.

При этом схему можно задать в виде

$$G(X \cup E \cup C, U),$$

где вершины  $X$  соответствуют элементам схемы,  $E$  – вершины графа, соответствующие цепям схемы,  $C$  – множество вершин графа, соответствующие контактам элементов, множество ребер

$$U = F \cup W,$$

где  $F$  – определяет принадлежность контактов из множества  $C$  элементам  $X$  и задается парами

$$f = (x_i, c_s);$$

ребра подмножества  $W$  задаются вхождением контакта из множества  $C$  в цепи  $E$  и описываются парами

$$w = (c_s, e_j);$$

Граф вида  $G(X \cup E \cup C, U)$  задается обычно в виде двух матриц  $A_1, A_2$ .

$$A_1 = \left\| a_{ij}^1 \right\|_{E \times C},$$

$$a_{ij}^1 = \begin{cases} 1, & \text{если } c_s \in e_j \\ 0, & \text{если - нет} \end{cases}$$

$$A_2 = \left\| a_{ij}^2 \right\|_{X \times C},$$

$$a_{ij}^2 = \begin{cases} 1, & \text{если } c_s \in x_i \\ 0, & \text{если - нет} \end{cases}$$

*Пример.* (см. рисунок 3.1)

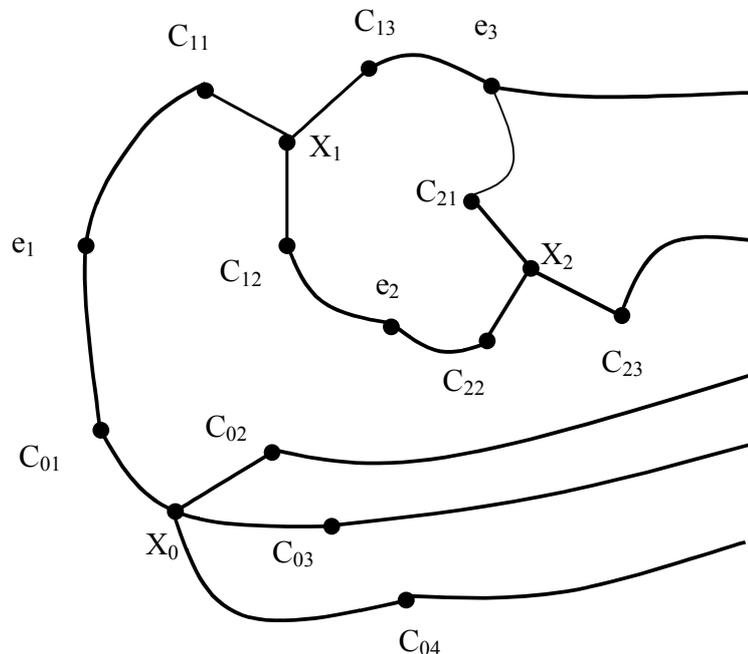


Рисунок 3.3 – Граф схемы электрической  $G$

Остальную часть графа  $G(X \cup E \cup C, U)$  доделать самим. В данном

графе вершины  $X+E+C$ , а ребра  $U=F+W$ .

0 условно не показаны:

	C <sub>01</sub>	C <sub>02</sub>	C <sub>03</sub>	C <sub>04</sub>	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>21</sub>	C <sub>22</sub>	C <sub>23</sub>	C <sub>31</sub>	C <sub>32</sub>	C <sub>33</sub>	C <sub>41</sub>	C <sub>42</sub>
$e_1$	1				1										
$e_2$						1			1						
$A_1 = e_3$							1	1				1			
$e_4$		1									1				
$e_5$			1							1			1	1	
$e_6$				1											1
	C <sub>01</sub>	C <sub>02</sub>	C <sub>03</sub>	C <sub>04</sub>	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>21</sub>	C <sub>22</sub>	C <sub>23</sub>	C <sub>31</sub>	C <sub>32</sub>	C <sub>33</sub>	C <sub>41</sub>	C <sub>42</sub>
$x_0$	1	1	1	1											
$x_1$					1	1	1								
$A_2 = x_2$								1	1	1					
$x_3$											1	1	1		
$x_4$														1	1

## 4 Алгоритмизация задач конструкторского проектирования

### 4.1 Покрытие функциональных схем модулями

При конструкторском проектировании ЭА решаются задачи, связанные с поиском наилучшего варианта конструкции, удовлетворяющего требованиям технического задания и максимально учитывающего возможности технологической базы производства. Тесная взаимосвязанность задач и большая размерность каждой из них обычно не позволяют предложить метод поиска оптимального конструктивного решения в едином цикле в связи с трудностями создания общей математической модели, комплексно учитывающей особенности конструкторско-технологической базы производства.

Существуют алгоритмы решения отдельных задач:

1. компоновки,
2. размещения,
3. трассировки.

**Компоновка** – процесс перехода от логическо-функционального или схемного описания устройства к конструктивному,

Т.е. это процесс распределения функциональных элементов схемы по группам, соответствующим модулям различного уровня.

Различают 3 типа задач компоновки:

- 1) Типизация (разбиение функциональной схемы на части, номенклатура которых известна)
- 2) Покрытие (преобразование функциональной схемы в схему соединений элементов, номенклатура которых известна)
- 3) Разрезание (разбиение ЭЗ на части, минимально связанные между собой)

**Математическая формулировка покрытия.** Исходными данными для решения задачи покрытия являются: *функциональная схема устройства и схемы типовых конструктивных элементов используемого набора модулей.*

Необходимо найти такое распределение логических функций покрываемой схемы по отдельным конструктивным элементам, при котором достигается экстремум целевой функции.

Известные в литературе алгоритмы покрытия оптимизируют в основном следующие показатели качества:

- 1) суммарную стоимость модулей, покрывающих схему;
- 2) общее число модулей, необходимое для реализации схемы;
- 3) число типов используемых модулей;
- 4) число межмодульных соединений и т.д.

Ограничениями обычно являются требования на совместную или раздель-

ную компоновку в едином конструктивном модуле элементов функциональной схемы, связанные с обеспечением нормального теплового режима, помехозащитности и простоты диагностики.

Наиболее распространенный вариант задачи - критерием качества является суммарная стоимость модулей. Для ее решения используют методы целочисленного линейного программирования.

Математически – необходимо минимизировать целевую функцию

$$F(X) = \min \sum_{i=1}^n C_i X_i$$

где  $C_i$  - стоимость  $i$ -го конструктивного модуля,  $X_i$  - число модулей  $i$ -го типа, которые необходимы для покрытия схемы.

*При ограничениях:*

- число логических функций любого  $i$ -го типа, входящих во все покрывающие модули, должно быть не меньше общего числа элементов соответствующего типа в реализуемой схеме;
- любой модуль используется только полностью, независимо от числа задействованных в нем компонентов.

Дополнительные условия качества могут усложнять целевую функцию (например, условие максимального расположения некоторых функций внутри некоторых модулей).

***В основу алгоритмов покрытия положены следующие этапы:***

- выделяют из заданной функциональной схемы подсхемы (группы максимально связанных между собой логических элементов),
- перебирают все или достаточно большое их число и проверяют на совпадение логических функций элементов подсхем и компонентов модулей используемого набора.
- Подсхема закрепляется за тем модулем, в состав которого входит наибольшее количество ее логических функций.
- Процесс продолжается до полного распределения элементов функциональной схемы по модулям.

Решение задачи покрытия, как правило, в 2 этапа:

1. Предварительного
2. Окончательного

**Алгоритм покрытия (эвристического типа):**

1. Из списка элементов схемы, подлежащей покрытию, выбирают очередной элемент  $m_i \in M$ .

Если список пуст – переходим к п. 9.

2. Определяем все логические элементы схемы из числа непокрытых и связанных с  $m_i$ .

3. Выбираем ячейки с числом выходов, равном числу взаимосвязанных логических элементов функциональной схемы.

4. Производим позиционное сравнение логических функций выбранных ячеек и логических функций взаимосвязанных элементов исходной функциональной схемы. Фиксируем все удовлетворительные варианты и переходим к п. 7.

Если ни одна ячейка не подходит переходим к п. 5

5. Уменьшаем число выбранных взаимосвязанных логических элементов исходной схемы на 1 и переходим к п. 3.

Если число выходов равно 1 – переходим к п. 6.

6. Выбираем все ячейки, имеющие один выход и производим позиционное сравнение логических функций ячейки и элементов исходной функциональной схемы. Фиксируем возможные варианты покрытия.

7. Выбираем ячейку с минимальной стоимостью.

8. Всем логическим элементам (выбранным) исходной функциональной схемы присваиваем порядковый номер в соответствии с номером ячейки. Переходим к п. 1.

9. Проводим ручную доработку покрытия.

Для улучшения полученного результата при окончательном покрытии осуществляют парные перестановки однотипных логических элементов различных ячеек. Критерием удачной перестановки является увеличение внутренних связей ячеек.

## **4.2 Компоновка конструктивных элементов по коммутационным платам**

Модульный метод проектирования предусматривает расчленение ЭА на отдельные конструктивно законченные единицы (модули) различных уровней: стойки, блоки, субблоки, функциональные узлы.

При разработке конструкции ЭА необходимо распределить элементы схемы (модулей предыдущего уровня) по коммутационным платам (коммутационным пространствам) данного уровня иерархии.

При ее решении основным критерием оптимальности компоновки модулей является минимизация числа межмодульных связей, что необходимо для

- повышения надежности схем (за счет уменьшения числа разъемных соединений),
- уменьшения влияния наводок и времени задержки сигнала в цепях (вследствие минимизации суммарной длины соединений),
- упрощения конструкции и
- повышения технологичности разрабатываемого устройства.

### Математическая формулировка.

Электрическую схему интерпретируют ненаправленным мультиграфом  $G(X, U)$ , в котором каждому конструктивному элементу (модулю) ставят в соответствие вершину мультиграфа  $X$ , а электрическим связям схемы - его ребра  $U$ . Требуется «разрезать» его на отдельные куски  $G_1(X_1, U_1)$ ,  $G_2(X_2, U_2)$ , ...,  $G_k(X_k, U_k)$  так, чтобы число ребер, соединяющих эти куски, было минимальным, т. е.

$$\min \sum_{i=1}^k \sum_{j=1}^k U_{ij}, i \neq j,$$

где  $U_{ij}$  - множество ребер, соединяющих куски  $G_i(X_i, U_i)$  и  $G_j(X_j, U_j)$ .

### Конструктивные ограничения в задачах компоновки:

- Число кусков разрезания графа  $k$ ;
- число вершин в каждом из кусков графа (определяется числом конструктивных элементов, которые необходимо разместить на коммутационной плате, пластине БИС, подложке БГИС и т. д.);
- максимальное число внешних связей каждого отдельно взятого куска графа (определяется количеством контактов используемого разъема, числом
- выводов стандартного корпуса БИС или БГИС);
- требование на раздельную компоновку отдельных вершин в различных кусках графа (обусловлено конструктивными соображениями и условиями электромагнитной совместимости).

### 4.3 Классификация алгоритмов компоновки

1. алгоритмы, использующие методы целочисленного программирования;
2. последовательные алгоритмы;
3. итерационные алгоритмы;
4. смешанные алгоритмы;
5. алгоритмы, основанные на методе ветвей и границ.

1) для устройства реальной сложности фактически не реализуемы на ЭВМ,  
2) При использовании **последовательных алгоритмов** сначала по определенному правилу выбирают первую вершину графа, затем осуществляют последовательный выбор вершин (из числа нераспределенных) и присоединение их к формируемому куску графа. После образования первого куска переходят ко второму и т. д. до получения желаемого разрезания исходного графа.

3) В **итерационных алгоритмах** начальное разрезание графа на куски вы-

полняют произвольным образом; оптимизация компоновки достигается парными или групповыми перестановками вершин графа из различных кусков. Процесс перераспределения вершин заканчивают при получении локального экстремума целевой функции, удовлетворяющего требованиям разработчика.

4) В *смешанных алгоритмах* компоновки для получения начального варианта «разрезания» используется алгоритм последовательного формирования кусков; дальнейшая оптимизация решения осуществляется перераспределением вершин между отдельными кусками графа.

5) *Метод ветвей и границ* сводится к последовательному разбиению множества допустимых планов задачи целочисленного программирования  $L$  на подмножества. Процесс разбиения продолжается до тех пор, пока каждое подмножество не будет представлять собой точку в многомерном пространстве. Процесс последовательного сечения исходного множества решений  $L$  гиперплоскостями представляет собой построение дерева ветвлений с соответствующими подмножествами. Чтобы уменьшить объем перебора, производят оценку образуемых подмножеств по наибольшему (наименьшему) значению для задач максимизации (минимизации) целевой функции. Это позволяет на каждом шаге исключать из рассмотрения те подмножества, для которых ожидаемое значение целевой функции оказывается наихудшим.

Чуть подробнее рассмотрим алгоритмы 1) и 2)

### **Последовательный алгоритм:**

«разрезание» исходного графа  $G(X, U)$  на куски  $G_i(X_i, U_i)$  сводится к следующему. В графе  $G(X, U)$  находят вершину  $x_i \in X$  с минимальной локальной степенью  $\rho$ :

$$\rho(x_i) = \min_{x_j \in X} \rho(x_j)$$

Число ребер инцидентных некоторой вершине  $x_i$  называется **степенью вершины**.

Из подмножества вершин, смежных с вершинами формируемого куска графа  $G_1(X_1, U_1)$ , выбирают ту, которая обеспечивает минимальное приращение связей куска с еще нераспределенными вершинами. Данную вершину  $x_j$  включают в  $G_1(X_1, U_1)$ , если не происходит нарушения ограничения по числу внешних связей куска.

Указанный процесс продолжается до тех пор, пока множество  $X_1$  не будет содержать  $n$  элементов либо присоединение очередной нераспределенной вершины  $x_j$ , к куску  $G_1(X_1, U_1)$  не приведет к нарушению ограничения по числу внешних соединений куска.

Процесс повторяют для формирования второго, третьего и т. д. кусков исходного графа с той лишь разницей, что рассмотрению подлежат вершины, не

вошедшие в предыдущие куски.

***Достоинства:***

- прост,
- легко реализуется на ЭВМ и
- позволяет быстро получить решение задачи компоновки.

Данный алгоритм может эффективно применяться и при наличии ограничения на совместную компоновку отдельных вершин графа. В этом случае каждая такая вершина жестко закрепляется за определенным куском графа и формирование очередного куска начинается с непосредственного выбора этой вершины в качестве исходной.

**Итерационный алгоритм:**

Сущность алгоритма заключается в выборе некоторого начального «разрезания» исходного графа на куски (вручную или с помощью последовательного метода компоновки) и последующего его улучшения с помощью итерационного парного или группового обмена вершин из различных кусков. При этом для каждой итерации осуществляется перестановка тех вершин, которая обеспечивает максимальное уменьшение числа связей между кусками графа или максимальное улучшение другого выбранного показателя качества с учетом используемых ограничений (например, на максимальное число внешних ребер любого отдельно взятого куска).

Парная перестановка вершин приведет к изменению числа только тех связывающих куски ребер, которые инцидентны этим вершинам.

Операция парного обмена вершин сводится к перестановке соответствующих строк и столбцов матрицы смежности  $A$ .

Процесс оптимального разрезания графа на куски заключается в отыскании на каждой итерации таких парных перестановок строк и столбцов матрицы  $A$ . При которых максимизируется сумма элементов в диагональных подматрицах, что равносильно минимизации числа соединительных ребер.

***Недостатки***

- Оптимальность решения в значительной степени зависит от того, насколько удачно было произведено начальное разделение графа.
- Требуют больших затрат машинного времени, чем последовательные алгоритмы.

***Достоинства:***

- Обеспечивают высокое качество решения задачи.
- Для сокращения числа итераций обмена вершин между кусками в смешанных алгоритмах для получения начального «разрезания» графа применяют последовательные методы формирования его кусков.

## 5 Алгоритмы размещения конструктивных модулей

### 5.1 Постановка задачи размещения конструктивных модулей различных уровней иерархии

После распределения конструктивных элементов (КЭ) ЭА в монтажном пространстве различного уровня иерархии, для каждой полученной в результате компоновки сборочной единицы производят размещение включенных в ее состав элементов предыдущего уровня, т.е. выбирают такое их взаимное расположение, при котором наилучшим образом учитываются предъявляемые к аппаратуре требования.

**Постановка задачи.** Исходными данными при решении задач размещения являются:

1. данные о конфигурации и размерах коммутационного пространства, (определяются требованиями установки и крепления данной сборочной единицы в аппаратуре);
2. количество и геометрические размеры конструктивных элементов, подлежащих размещению;
3. принципиальная схема узла (или КЭ);
4. ограничения на взаимное расположение отдельных элементов, учитывающих особенности разрабатываемой конструкции.

Задача размещения сводится к отысканию для каждого размещаемого элемента таких позиций, при которых оптимизируется выбранный показатель качества и обеспечиваются наиболее благоприятные условия для последующего электрического монтажа.

Особое значение эта задача приобретает при проектировании аппаратуры на печатных платах.

Основная сложность в постановке задач размещения - выбор целевой функции. Связано это с тем, что одной из главных целей размещения является создание наилучших условий для дальнейшей трассировки соединений, что невозможно проверить без проведения самой трассировки.

Любые другие способы оценки качества размещения (минимум числа пересечений ребер графа, интерпретирующей электрическую схему соединений, разбиение графа на минимальное число плоских суграфов и т. д.), хотя и позволяют создать благоприятные для трассировки условия, но не гарантируют получение оптимального результата.

Если для оценки качества размещения элементов выбрать критерий, непосредственно связанный с получением оптимального рисунка металлизации печатной платы, то конечный результат может быть найден только при совместном решении задач размещения, выбора очередности проведения соединений и трассировки, что трудно осуществить вследствие огромных затрат машинного

времени.

Поэтому все применяемые в настоящее время алгоритмы размещения используют *промежуточные критерии*, которые лишь качественно способствуют решению основной задачи: *получению оптимальной трассировки соединений*.

- 1) *минимум суммарной взвешенной длины соединений;*
- 2) минимум числа соединений, длина которых больше заданной;
- 3) минимум числа пересечений проводников;
- 4) максимальное число соединений между элементами, находящимися в соседних позициях либо в позициях, указанных разработчиком;
- 5) максимум числа цепей простой конфигурации.

Наибольшее распространение в алгоритмах размещения получил первый критерий, что объясняется следующими причинами: уменьшение длин соединений

- улучшает электрические характеристики устройства,
- упрощает трассировку печатных проводников и
- снижает трудоемкость изготовления печатных плат;
- сравнительно прост в реализации.

При практической реализации алгоритмов размещения часто используют представление конструктивных элементов и позиций на коммутационной плате точками, совпадающими с их геометрическими центрами, а все соединения между элементами приводят к попарно взвешенным связям. Весовые оценки связей учитывают такие характеристики схемы, как

- число электрических цепей между элементами,
- теплонагруженность элементов,
- условия распространения сигналов в цепях и т. д.

Широкий класс таких оценок описывается формулой, определяющей значение коэффициента взвешенной связности:

$$c_{ij} = \sum_{s=1}^g a_{ij}^s T_{ij}^s ,$$

где  $T_{ij}^s$  - вес s-той цепи, связывающей элементы  $i$  и  $j$ .

$a_{ij}^s$  - элемент матрицы смежности между элементами  $i$  и  $j$  по s-той цепи.

$g$  – число цепей, по которым связаны элементы  $i$  и  $j$ .

$T_{ij}^s$  определяет важность s-той цепи с точки зрения минимизации ее длины с учетом условий распространения сигнала, теплонагруженности, электромагнитной совместимости и т.д.

В общем виде задача размещения конструктивных элементов на коммутационной плате формулируется следующим образом. Задано множество конструктивных элементов  $X$  и множество связей между этими элементами  $U$ , а также множество установочных мест (позиций) на коммутационной плате  $T$ . Найти такое отображение множества  $X$  на множество  $T$ , которое обеспечивает экстремум целевой функции  $F$ .

Если критерием качества размещения является минимум суммарной взвешенной длины соединений, то задача состоит в минимизации функции

$$F = \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{ij}$$

Обычно поле позиций (коммутационная плата) имеет форму прямоугольника  $axb$  с координатами  $0 \leq x \leq a$  и  $0 \leq y \leq b$ .

Вся площадь платы разбивается на ряд областей (позиций), число которых должно быть не меньше числа размещаемых элементов. В результате получим фиксированные позиции для установки элементов.

Исходя из конструктивных соображений, могут выделять области для размещения выводных контактных зон схемы (разъемов), а также запрещенные области, в которых не должны помещаться элементы схемы (зоны крепления платы, конструктивные вырезы и т. п.).

Все конструктивные элементы, подлежащие размещению, можно условно разделить на три группы

1) *нефиксированные элементы*, местоположение которых на плате заранее не известно.

2) *граничные элементы*, к которым относятся элементы, связанные с разъемами, осуществляющими электрическую связь с элементами, расположенными на других коммутационных платах. Так как разъемы обычно помещают на внешней стороне коммутационной платы, то эти элементы желательно располагать у границы коммутационного поля.

3) *фиксированные элементы*, местоположение которых на плате заранее определено (указано разработчиком).

Оптимизацию размещения осуществляют, как правило, локально из некоторого случайного или интуитивно выбранного первоначального размещения.

## 5.2 Классификация алгоритмов размещения

По принципам реализации известные алгоритмы размещения можно разделить на алгоритмы, использующие непрерывно-дискретные и дискретные методы оптимизации (рисунок 5.1). Эффективность того или иного алгоритма обычно оценивают по результатам решения типовых конструкторских задач.

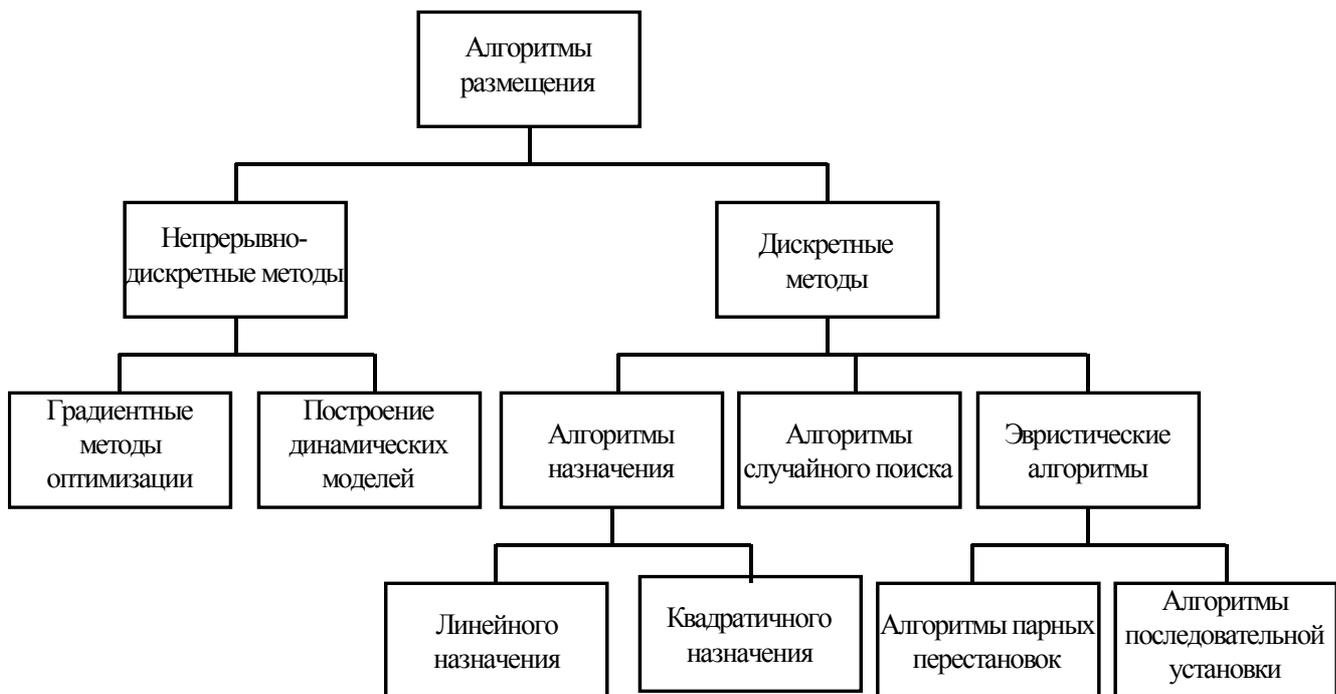


Рисунок 5.1 – Классификация алгоритмов размещения

### 5.3 Непрерывно-дискретные методы размещения

Для них задание фиксированного набора посадочных мест не обязательно. Размещение элементов осуществляется на непрерывной плоскости.

Представляют наибольший интерес для конструкций, содержащих разногабаритные элементы (ППП + гибридные ИМС + дискретные элементы).

При использовании данных методов оптимизации, задача размещения решается в два этапа:

- 1) определяют координаты местоположения центров элементов, при которых целевая функция  $F$  имеет экстремальное значение;
- 2) полученные координаты «округляются» в фиксированные целочисленные значения координатной сетки, нанесенной на поверхность коммутационной платы.

#### Алгоритмы, использующие градиентные методы

Получили наибольшее развитие.

При использовании градиентных методов для отыскания оптимального размещения элементов на плате по критерию минимума суммарной взвешенной длины соединений решение задачи сводится к минимизации целевой функции  $F$ . Так как целевая функция является многомерной, то градиент аналитически выражают в виде суммы частных производных по всем нефиксированным координатам  $x_i$  и  $y_i$ :

$$\nabla F = \sum_{i=1}^q \frac{\partial F}{\partial x_i} x_i + \sum_{i=1}^h \frac{\partial F}{\partial y_i} y_i$$

Для нахождения минимального значения целевой функции  $F$  используют пошаговый градиентный метод, подробно рассмотренный ранее. Движение по координатам осуществляют до тех пор, пока на очередной итерации частные производные не будут меньше некоторой фиксированной величины (погрешности).

При использовании градиентных методов местоположение конструктивных элементов на плате получается в непрерывных координатах. Поэтому на втором этапе проводят «округление» полученных величин до ближайших целочисленных координат, т.е. все нефиксированные элементы сдвигаются в фиксированные позиции координатной сетки платы.

#### ***Достоинства:***

- 1) сравнительно небольшие затраты машинного времени на отыскание экстремума целевой функции,
- 2) наличие стандартных программ для решения данного класса задач

#### ***Недостатки:***

- 1) возможность получения лишь локального экстремума;
- 2) низкая эффективность при пологом экстремуме, так как раньше времени приходим к условному решению;
- 3) большая неравномерность распределения элементов на плате до «округления» координат.

В связи с указанными недостатками в чистом виде градиентные методы для решения задач размещения не используют. Их применяют в сочетании с методами случайного поиска, отыскивая при этом глобальный экстремум целевой функции.

### **Алгоритмы, использующие динамические модели**

Процесс размещения элементов на плате представляется механической моделью.

Элементы считаются материальными точками, на каждую из которых действуют силы притяжения и отталкивания.

Силы притяжения, действующие между любыми двумя материальными точками  $x_i$  и  $x_j$  пропорциональны числу электрических связей между данными конструктивными элементами/

Состояние равновесия такой системы соответствует минимуму суммарной длины всех соединений.

Введение сил отталкивания материальных точек друг от друга и от границ платы исключает возможность слияния двух любых точек и способствует их равномерному распределению по поверхности монтажного поля.

Чтобы устранить возникновение в системе незатухающих колебаний, вводят силы сопротивления среды, пропорциональные скорости движения материальных точек.

Задача оптимального размещения элементов сводится к нахождению такого местоположения точек, при котором равнодействующие всех сил обращаются в нуль.

Решение задачи осуществляют в три этапа.

1) используя критерий минимума суммарной взвешенной длины связей, производят размещение материальных точек на условном поле позиций без учета требования равномерности их распределения по поверхности и попадания точек в фиксированные позиции поля. Полученное решение используют в качестве начального размещения на втором этапе,

2) на материальные точки начинают действовать силы притяжения и отталкивания. Под влиянием этих сил точки начинают перемещаться к положению равновесия системы, при котором обеспечивается приемлемая степень равномерности их размещения на поле позиций.

3) точки сдвигаются в фиксированные позиции платы при минимально возможных изменениях их взаиморасположения.

#### ***Достоинства:***

- 1) возможность получения глобального экстремума целевой функции,
- 2) сведение поиска к вычислительным процедурам, для которых имеются разработанные численные методы.

#### ***Недостатки:***

- 1) трудоемкость метода и сложность его реализации (подбора коэффициентов для силовых связей);
- 2) необходимость фиксирования местоположения некоторого числа конструктивных элементов на плате для предотвращения большой неравномерности их размещения на отдельных участках платы.

Метод целесообразно использовать в тех случаях, когда позиции для установки элементов на коммутационной плате заранее не фиксированы, что имеет место при размещении разногабаритных элементов (печатные платы аналого-цифровой техники, топология микросхем и т.п.).

## 5.4 Алгоритмы, использующие дискретные методы оптимизации

В дискретных методах оптимизации модель коммутационного пространства представляют в виде множества фиксированных координат позиций. Задача размещения сводится к сравнению различных вариантов, закрепления элементов в этих позициях и выбору того из них, который обеспечивает экстремальное значение целевой функции  $F$ .

В этих условиях для нахождения глобального экстремума  $F$  необходим полный перебор всех возможных вариантов размещения, т. е. для оптимального размещения  $n$  элементов в  $k$  позициях следует осуществить  $C_k^n n!$  перестановок, что уже при  $n = 10$  практически невозможно, так как минимальное число вариантов размещения ( $k = n$ ) оказывается равным 3 628 000. Поэтому дискретные методы оптимизации позволяют отыскивать обычно только локальные экстремумы целевой функции.

### Алгоритмы назначения

По основным принципам построения алгоритмы **назначения** делят на две группы.

1. Алгоритмы линейного назначения
2. Алгоритмы квадратичного назначения

#### 1) Алгоритмы линейного назначения

Основаны на комбинаторно-аналитическом алгоритме Штейнберга, в котором объектами рассмотрения являются несвязные подмножества конструктивных элементов, т.е. такие подмножества, для которых матрица связности  $S$  полностью состоит из нулей.

Суть метода Штейнберга:

Пусть имеется некоторое начальное размещение конструктивных элементов. По матрице связности  $S$  выделяем максимальное внутренне устойчивое подмножество  $R_i$ , т. е. максимальную совокупность несвязных между собой элементов.

Из начального размещения исключаем элементы, принадлежащие  $R_i$ .

Для всех элементов из подмножества ищем такие позиции из числа свободных, которые соответствуют минимуму целевой функции  $F$ .

Так как элементы подмножества  $R_i$  не связаны друг с другом, то на выбор позиции для любого  $r_j \in R_i$  оказывают влияние только связи  $r_j$  с элементами подмножества  $R \setminus R_i$  (мн-во  $R$  за исключением  $n/мн$ -ва  $R_i$ ), где  $R$  - множество конструктивных элементов, размещаемых на плате.

Таким образом, оптимизация размещения элементов подмножества  $R_i$  на плате сводится к решению задачи о назначениях (см. линейное программиро-

вание).

После нахождения оптимального размещения всех элементов  $r_j \in R_i$  выделяем следующее внутренне устойчивое подмножество и т.д.

Рассмотренный итерационный процесс оптимизации размещения приводит к монотонному уменьшению значения целевой функции  $F$ . Условием окончания поиска на  $z$ -ом шаге является незначительное уменьшение целевой функции при оптимизации размещения очередного внутренне устойчивого подмножества:

$$\frac{F^{z-1} - F^z}{F^{z-1}} \leq \varepsilon, \quad (5.1)$$

где  $F^{z-1}$  и  $F^z$  - значения целевой функции на  $(z-1)$ - и  $z$ -м шагах;  $\varepsilon$  - порог чувствительности алгоритма.

Эффективность алгоритма Штейнберга зависит от числа нулей в матрице связности  $S$ . Обычно такая матрица содержит более 60% нулей, вследствие чего процесс поиска быстро сходится.

***Недостатки:***

1. большой объем требуемой памяти ЭВМ
2. возможность нахождения только локального экстремума целевой функции (глобальный оптимум на каждом шаге ищется лишь для внутренне устойчивого подмножества, а не для всей системы в целом).

**2) Алгоритмы квадратичного назначения**

Основаны на использовании методов нелинейного программирования.

Наибольшее распространение получили алгоритмы, основанные на методе ветвей и границ.

***Достоинства:***

1. возможность получения глобального экстремума целевой функции,
2. наличие типовых программ решения задач квадратичного назначения.

***Недостатки:***

1. необходимость качественного начального размещения
2. сравнительно большие затраты машинного времени, не позволяющие решать задачи большой размерности.

**Алгоритмы случайного поиска**

По основным принципам построения алгоритмы **случайного поиска** делят на три группы.

1. Алгоритмы слепого поиска

2. Алгоритмы случайного блуждания
3. Комбинированные алгоритмы случайного поиска

### 1) Алгоритмы слепого поиска.

Выбирают наугад какую-либо позицию монтажной плоскости из числа незанятых и на ней закрепляют (по порядку, начиная с первого) подлежащий размещению элемент. Эту операцию продолжают до тех пор, пока все конструктивные элементы не будут установлены. Для полученного размещения вычисляют значение целевой функции, например суммарную взвешенную длину соединений.

Аналогично проводят второе, третье и т. д. случайные размещения элементов, начиная с закрепления второго, третьего и т. д. элементов. Подобная процедура обеспечивает равновероятное попадание элементов на различные позиции и осуществляется с помощью датчика псевдослучайных чисел, равномерно распределенных на отрезке  $[-1, +1]$ .

Вычисленное для каждого вновь найденного варианта размещения значение целевой функции сравнивают с наилучшим результатом, достигнутым на предыдущих шагах. Если имеет место улучшение значения целевой функции, то данное размещение запоминают, в противном случае — отбрасывают как неудачное.

#### *Достоинство:*

Алгоритмы не накладывают никаких ограничений на свойства области допустимых значений параметров и целевую функцию, позволяя проводить оптимизацию одновременно по нескольким показателям качества.

#### *Недостаток:*

Для нахождения глобального экстремума требуется просмотреть огромное число вариантов размещения, практически равное полному регулярному перебору.

Число просматриваемых вариантов  $q$  можно сократить, если отыскивать не оптимальное, а близкое к нему размещение, для которого значение целевой функции  $F$  отличается от оптимального  $F^*$  на величину, не превосходящую некоторую заранее заданную погрешность  $\varepsilon$ . Поиск ведется до тех пор, пока на  $q$ -ом шаге не будет выполнено условие

$$\left| F^* - F_q \right| \leq \varepsilon, \quad (5.2)$$

где  $F_q$  - наилучшее значение целевой функции, достигнутое за  $q$  шагов.

Приближенное значение  $F^*$  можно определить следующим образом. В процессе работы алгоритма накапливается информация о значениях  $F_i$ ,  $i = 1, 2, \dots, z$  ( $z \leq q$ ). На  $z$ -ом шаге, экстраполировав монотонно изменяющуюся последовательность значений  $F_i$ , находят предел  $F_e$ , к которому стремится целевая функция. Это значение и принимается за искомое  $F^*$ . При большом  $z$  данный метод обеспечивает достаточно высокую точность оценки  $F^*$ .

## **2) Алгоритмы случайного блуждания.**

Позволяют снизить трудоемкость поиска оптимального решения вследствие учета характерных особенностей оптимизируемой функции. Например, можно воспользоваться тем обстоятельством, что при оптимальном расположении элементов на плате вероятность установки сильно связанных между собой элементов в соседние позиции выше, чем в удаленные.

Для повышения вероятности попадания связанных элементов в соседние позиции перед каждой очередной установкой элемента выявляют те позиции, в которых располагаются связанные с ним элементы. Все свободные позиции нумеруются в порядке возрастания их расстояния от данных позиций. С помощью датчика псевдослучайных чисел, распределенных в интервале  $[0, + 1]$  с максимальной плотностью вблизи нуля, осуществляют случайный выбор позиции таким образом, чтобы наиболее вероятным было попадание устанавливаемого элемента в позицию с меньшим номером, после чего производят перенумерацию оставшихся незанятых позиций, принимая во внимание связи следующего закрепляемого элемента и т.д.

В остальном данный алгоритм не отличается от предыдущего.

## **3) Комбинированные алгоритмы случайного поиска.**

### ***Суть:***

Представляют собой комбинацию метода случайного поиска и какого-либо регулярного метода.

### ***Достоинства:***

1. простота учета конкретных конструкторско-технологических ограничений,
2. возможность проводить оптимизацию одновременно по нескольким показателям качества,
3. наличие типовых программ, обеспечивающих случайный поиск.

### ***Недостатки:***

1. требование одинаковых геометрических размеров размещаемых элементов (условие регулярности)
2. быстрый рост затрат машинного времени при повышении точности нахождения глобального экстремума целевой функции.

## **5.5 Эвристические алгоритмы**

### ***Достоинства:***

1. Эвристические алгоритмы позволяют сократить время решения задачи при вполне приемлемом для практики качестве получаемого результата.
2. Они лучше приспособлены для учета конкретных конструкторско-технологических ограничений.

К данной группе алгоритмов относятся

1. итерационные,
2. последовательные и
3. смешанные (начальное размещение получают с помощью последовательного закрепления элементов в позициях, а улучшение размещения достигается в результате их парных перестановок).

#### 1) итерационные алгоритмы

К ним относятся

*а) алгоритмы парных перестановок,*

*б) алгоритмы групповых перестановок*

*в) алгоритмы последовательной установки*

*г) параллельные алгоритмы на основе метода обратного размещения*

### Структура итерационных алгоритмов

Алгоритмы данной группы используют общие идеи последовательных приближений и являются комбинаторными аналогами градиентных методов оптимизации.

В любом итерационном алгоритме исследуется некоторое подмножество размещений, близких к начальному, для выделения в нем размещения с минимальным значением целевой функции  $F$ .

Найденное значение вновь принимается за начальное и процесс повторяется.

Итерационные алгоритмы имеют сходную структуру и содержат этапы:

1. Преобразование очередного размещения.
2. Вычисление целевой функции размещения
3. Выбор наилучшего варианта размещения по п. 2.
4. Переход к следующей итерации и правило остановки.

Как правило, итерационный процесс заканчивается, как только разность значений целевой функции между  $(z-1)$  и  $z$ -й итерациями не превосходит некоторой заранее заданной величины  $\varepsilon$ :

$$\frac{F^{z-1} - F^z}{F^{z-1}} \leq \varepsilon . \quad (5.3)$$

Для итерационных алгоритмов характерно неравенство:

$$F^{нач} > F^{(1)} > F^{(2)} > \dots$$

### Алгоритмы парных перестановок

Имеется 2 разновидности, которые различаются по способу конструктивных элементов, подлежащих обмену.

1) Содержательно описывается следующим образом:

а) выбирают первый по порядку КЭ,

б) меняют его местами со всеми остальными, рассчитывая для всех вари-

антов значение показателя качества.

в) сравнивают полученные результаты, включая и исходное размещение.

г) в качестве исходной принимают ту схему размещения, которой соответствует наилучшее значение целевой функции.

д) после того, как найдена наилучшая перестановка для 1-го модуля, аналогичную процедуру выполняют для 2. 3 ... КЭ.

**Достоинства:**

1. алгоритм обладает быстрой сходимостью
2. алгоритм прост в программировании

2) Для отыскания оптимального размещения элементов при данном алгоритме производится перестановка  $i$ -й пары элементов, для которой уменьшается значение целевой функции.

Математически алгоритмы парных перестановок можно обосновать следующим образом:

пусть  $d_{ij}$  – матрица расстояний,

$c_{ij}$  – матрица связи,

тогда суммарная длина всех связей между элементами  $i$  и  $j$  равна:

$$l_{ij} = d_{ij} c_{ij}$$

для конкретных  $i$  и  $j$ .

Рассматривая это выражение на все  $n$  элементов, подлежащих размещению, получим:

$$L^{(i)} = \sum_{j=1}^n d_{ij} c_{ij}$$

Если выделить из множества элементы  $i$  и  $j$ , то суммарная длина связей этих элементов со всеми остальными определится:

$$L^{(i,j)} = L^{(i)} - 2d_{ij}c_{ij} = \sum_{t=1}^n (d_{it}c_{it} + d_{jt}c_{jt}) - 2d_{ij}c_{ij}$$

При попарной перестановке  $i$  и  $j$  элементов их суммарная длина со всеми остальными элементами станет равной

$$L^{(i,j)} = \sum_{t=1}^n (d_{it}c_{jt} + d_{jt}c_{it})$$

Член  $2d_{ij}c_{ij}$  в данном выражении отсутствует, т.к. при

$$t = i, t = j$$

элементы

$$d_{it} = 0, d_{jt} = 0$$

Изменение суммарной длины связи между КЭ  $i$  и  $j$  со всеми остальными, полученное при их перестановке будет равно:

$$\Delta L = L^{(j,i)} - L^{(i,j)} = 2d_{ij}c_{ij} - \sum_{t=1}^n (d_{it} - d_{jt})(c_{it} - c_{jt})$$

На основании этой формулы можно определить матрицу приращения и находить пары, перестановка которых уменьшает суммарную длину связи между КЭ.

Размещение элементов заканчивается, если в матрице приращений не имеется отрицательных чисел.

### **Алгоритмы групповых перестановок**

Возможен не только обмен двух КЭ, но и целых групп элементов.

#### ***Недостаток:***

сложность вычисления приращений целевой функции не окупается точностью полученного размещения.

В связи с этим такие алгоритмы не нашли широкого применения.

Так по экспертным данным эффект от циклических перестановок 3-х элементов и размещения полученного по алгоритму парных перестановок составляет всего лишь несколько процентов.

#### ***Достоинства:***

перспективно использовать для улучшения качества размещения, полученного другими способами.

### **Алгоритмы последовательной установки**

Основаны на допущении что для получения оптимального размещения необходимо в соседних позициях располагать элементы, максимально связанные друг с другом.

*Сущность:* в последовательном закреплении заданного набора конструктивных элементов на коммутационной плате относительно ранее установленных. В качестве первоначально закрепленных на плате элементов обычно выбирают разъемы, которые искусственно «раздвигают» до краев платы

При этом все контакты разъемов равномерно распределяются по секциям (столбцам и строкам координатной сетки). На каждом  $s$ -ом шаге для установки на коммутационную плату выбирают элемент  $x_i^s$  из числа еще неразмещенных,

имеющий максимальную степень связности с ранее закрепленными элементами  $R_{s-1}$ . В большинстве используемых в настоящее время алгоритмов оценку степени связности производят по одной из следующих формул:

$$\Phi_i^{(s)} = \sum_{j \in J_{s-1}} c_{ij}$$

$$\Phi_i^{(s)} = 2 \sum_{j \in J_{s-1}} c_{ij} - \sum_{j=1}^n c_{ij}$$

где  $c_{ij}$  - коэффициент взвешенной связности элементов  $i$  и  $j$ ;  $J_{s-1}$  - множество индексов элементов, закрепленных на предыдущих  $s-1$  шагах;  $n$  - общее число размещаемых элементов.

Если установочные размеры всех размещаемых на плате элементов одинаковы, то выбранный на очередном шаге элемент  $r_i^s$  закрепляют в той позиции  $t_j^s$  из числа незанятых, для которой значение целевой функции  $F_j^s$  с учетом ранее размещенных элементов  $R_{s-1}$  минимально.

В частности, если критерием оптимальности является минимум суммарной взвешенной длины соединений, то

$$F_j^s = \min_{t_f \in T \setminus T_{s-1}} \sum_{j \in J_{s-1}} c_{ij} d_{ff}$$

где  $d_{ff}$  - расстояние между  $f$ -й позицией установки элемента  $r_i^s$  и позицией размещенного ранее элемента  $r_j$ ;  $T_{s-1}$  - множество позиций, занятых элементами после  $(s-1)$ -го шага алгоритма.

Процесс размещения элементов заканчивается после выполнения  $n$  шагов алгоритма.

Рассмотренный алгоритм можно использовать и для размещения разногабаритных элементов, если их геометрические размеры кратны или близки к кратным. При этом поверхность платы условно покрывается прямоугольной координатной сеткой, линейные размеры ячеек которой равны соответствующим линейным размерам позиций минимальных КЭ.

Для повышения эффективности в алгоритме предусмотрена проверка целесообразности удаления ранее размещенных простых конструктивных элементов, препятствующих закреплению на плате более сложных.

Так при установке элемента  $u$  (состоящего из 4 минимальных) в зоне его предполагаемого закрепления просматриваются состояния четырех элементарных ячеек (свободны они или заняты), в совокупности представляющих собой установочное место этого элемента. Если эти ячейки свободны, то элемент  $u$  закрепляют, в противном случае оценивают возможность перемещения ранее закрепленных в данных ячейках минимальных элементов на новые позиции платы.

После размещения на коммутационной плате всех элементов полученное решение может быть улучшено за счет парных перестановок в пределах каждого класса (типоразмера).

***Достоинство:***

Данные алгоритмы являются в настоящее время самыми быстро действующими.

***Недостаток:***

по качеству получаемого решения – хуже других итерационных.

**Параллельные алгоритмы на основе метода обратного размещения**

В методе обратного размещения выполняется предварительная оценка каждого размещенного элемента  $x_i$  и каждого места печатной платы  $t_j$ .

После этого элементы размещаются одновременно.

Пусть заданы матрица связей и длин:

$$C = \left\| c_{ij} \right\|_{n \times n}$$
$$D = \left\| d_{ij} \right\|_{m \times m}$$
$$m \geq n$$

( $C$  – элементы,  $D$  – места на печатной плате).

Для каждого элемента  $x_i$  по матрицам  $C$  и  $D$  находим суммарное число связей этого элемента с остальными:

$$c_i = \sum_{j=1}^n c_{ij},$$
$$i = 1, 2, \dots, n$$

$$d_i = \sum_{j=1}^m d_{ij},$$

$$i = 1, 2 \dots m$$

$d_i$  – определяет суммарное расстояние от  $i$ -й позиции до всех остальных позиций.

Позиции в центральной части платы имеют меньшее  $d_i$  чем на периферии, поэтому центральные позиции наиболее благоприятны для размещения элементов с большим значением  $c_i$ . В связи с этим

**Алгоритм размещения:**

1. Упорядочивают элементы по возрастанию характеристики  $c_i$
2. Упорядочивают места печатной платы по убыванию характеристики  $d_i$
3. Определяется размещение, где каждый соответствующий элемент  $c_i$  закрепляется за соответствующим местом  $d_i$

**Пример.**

Задана монтажная плата

t1	t2	t3	t4	t5	t6
----	----	----	----	----	----

и матрицы связей и длин

$$C = \begin{vmatrix} 0 & 2 & 2 & 8 & 4 & 4 \\ 2 & 0 & 1 & 10 & 3 & 8 \\ 2 & 1 & 0 & 3 & 4 & 7 \\ 3 & 10 & 3 & 0 & 7 & 8 \\ 9 & 5 & 4 & 7 & 0 & 11 \\ 4 & 6 & 7 & 8 & 11 & 0 \end{vmatrix} \quad D = \begin{vmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 1 & 2 \\ 4 & 3 & 2 & 1 & 0 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \end{vmatrix}$$

$$C \Rightarrow \begin{aligned} C_1 &= 20 \\ C_2 &= 24 \\ C_3 &= 17 \\ C_4 &= 31 \\ C_5 &= 36 \\ C_6 &= 36 \end{aligned} \quad D \Rightarrow \begin{aligned} D_1 &= 15 \\ D_2 &= 11 \\ D_3 &= 9 \\ D_4 &= 9 \\ D_5 &= 11 \\ D_6 &= 15 \end{aligned}$$

1)  $C_i = 3, 1, 2, 4, 5, 6$

2)  $D_i = 1, 6, 2, 5, 3, 4$

3) Размещаем 3 элемент в 1 ячейке,  $1 \rightarrow 6, 2 \rightarrow 2, 4 \rightarrow 5, 5 \rightarrow 3, 6 \rightarrow 4$ .

Данный алгоритм прост и может использоваться даже при ручном размещении.

## **5.6 Особенности алгоритмов размещения при многоцелевой оптимизации модулей**

Особенностью оптимального конструирования ЭА является трудность установления единого критерия качества, приводящего к действительно оптимальной конструкции.

Например, при решении задач размещения элементов на печатных платах нужно не только учесть возможность последующей реализации печатного монтажа в соответствии с применяемой технологией, но и обеспечить нормальный тепловой режим и надежность создаваемой конструкции при минимальных затратах на ее производство и эксплуатацию. Кроме того, часто следует учитывать еще такие показатели качества, как коэффициент заполнения монтажной плоскости, ремонтпригодность и целый ряд специальных требований, связанных с особенностями разрабатываемой аппаратуры. Все это привело к необходимости создания методов многоцелевой оптимизации, среди которых наибольшее распространение получили следующие методы:

- 1) использования единого функционала;**
- 2) выбора ведущего показателя;**
- 3) параллельной оптимизации по нескольким показателям.**

### **1) Метод использования единого функционала**

Используется единый функционал  $F$ , но каждому показателю качества  $F_i$  свой весовой коэффициент  $k_i$  учитывающий его важность, а общий показатель эффективности представляют в виде взвешенной суммы (произведения) отдельных показателей.

#### ***Достоинство:***

Возможность варьирования весовыми коэффициентами и простота реализации на ЭВМ.

#### ***Недостаток:***

Трудность обоснования важности каждого показателя и значения конкретного значения весового коэффициента.

### **2) Метод выбора ведущего показателя**

Основан на использовании принципа последовательной субоптимизации результатов, получаемых на каждом этапе поиска. Все показатели качества располагают в порядке важности и сначала отыскивают оптимальное решение по первому из них. Остальные показатели выступают в роли ограничений, затем определяют допустимую область в которой значение первого показателя отличается от оптимального на некоторую величину  $\varepsilon$  (например, на 5-10%) и в этой области ищут оптимальное решение по второму показателю и т.д.

***Достоинство:***

Возможность учета в виде списка ограничений большого числа различных требований, предъявляемых к конструкции ЭА.

***Недостаток:***

Быстрый рост затрат машинного времени и объема памяти при расширении списка ограничений.

**3) Метод параллельной оптимизации по нескольким показателям**

Заключается в оценке различных вариантов размещения одновременно по всем оптимизируемым показателям.

Качество полученного размещения оценивается с помощью функционала:

$$L = \sum_{i=1}^n k_i \frac{F_i^0 - F_i^s}{F_i^0}$$

где  $k_i$  — коэффициент, учитывающий важность  $i$ -го показателя,  $n$  — число показателей,  $F_i^s$ ,  $F_i^0$  — соответственно, значение текущего показателя качества и базового.

Если  $L > 0$ , то новое размещение считают лучшим и принимают за базовое, в противном случае его отбрасывают как неудачное.

На первом шаге в качестве базовых значений показателей качества выбирают заведомо большие числа (из области допустимых), поэтому при оценке первого варианта размещения функционал  $L > 0$ . Процесс поиска продолжается до тех пор, пока значения функционала  $L$ , соответствующие выбору нового базового размещения, на протяжении большого числа шагов (20—30) не будут меньше некоторой наперед заданной погрешности  $\varepsilon$  или пока не будет исчерпан лимит времени на решение задачи.

***Достоинство:***

Возможность получения действительно оптимального решения по всем выбранным критериям качества.

***Недостаток:***

Большие затраты машинного времени.

6.1 Классификация алгоритмов трассировки

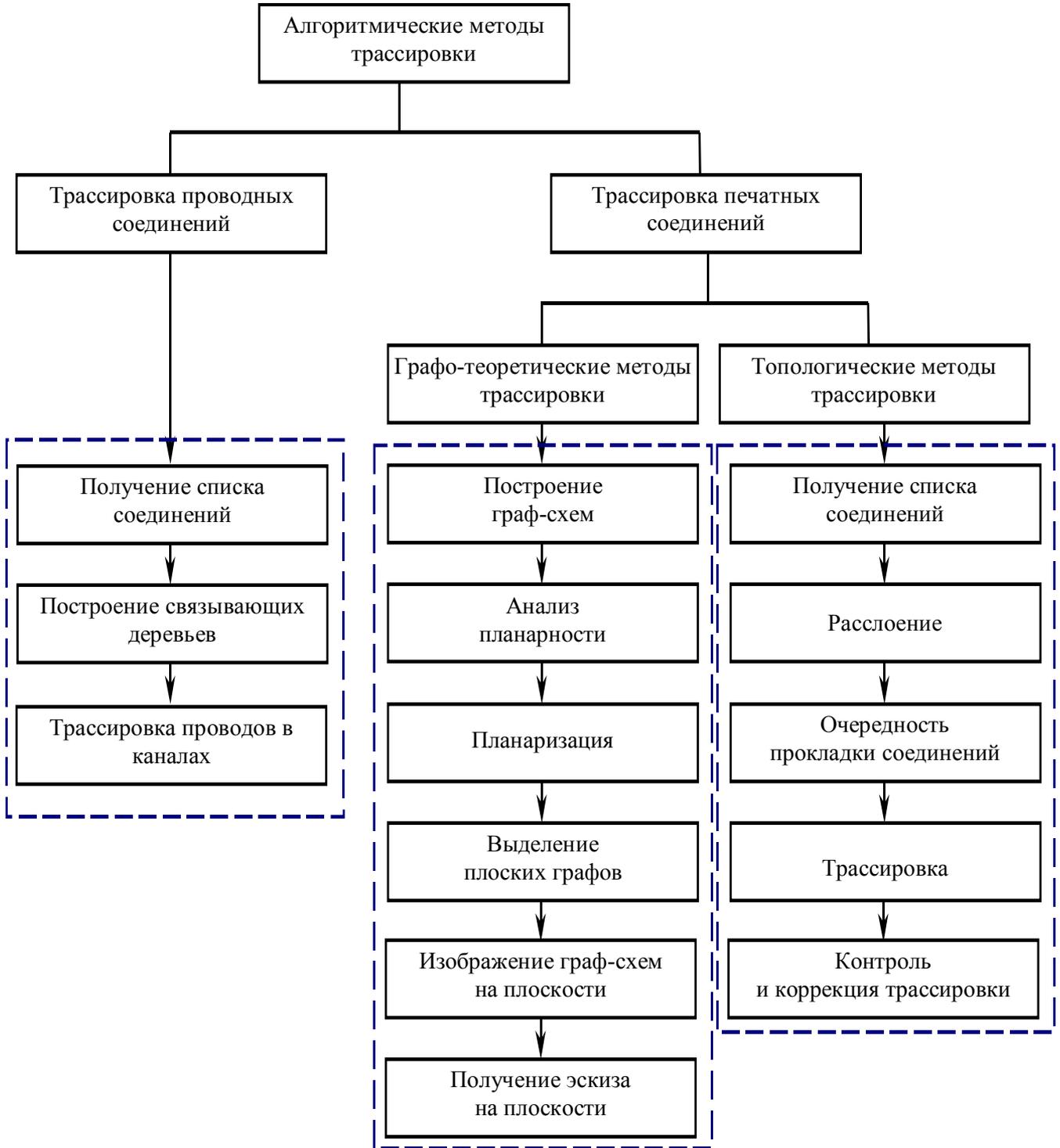


Рисунок 6.1 – Классификация алгоритмов трассировки (часть 1)

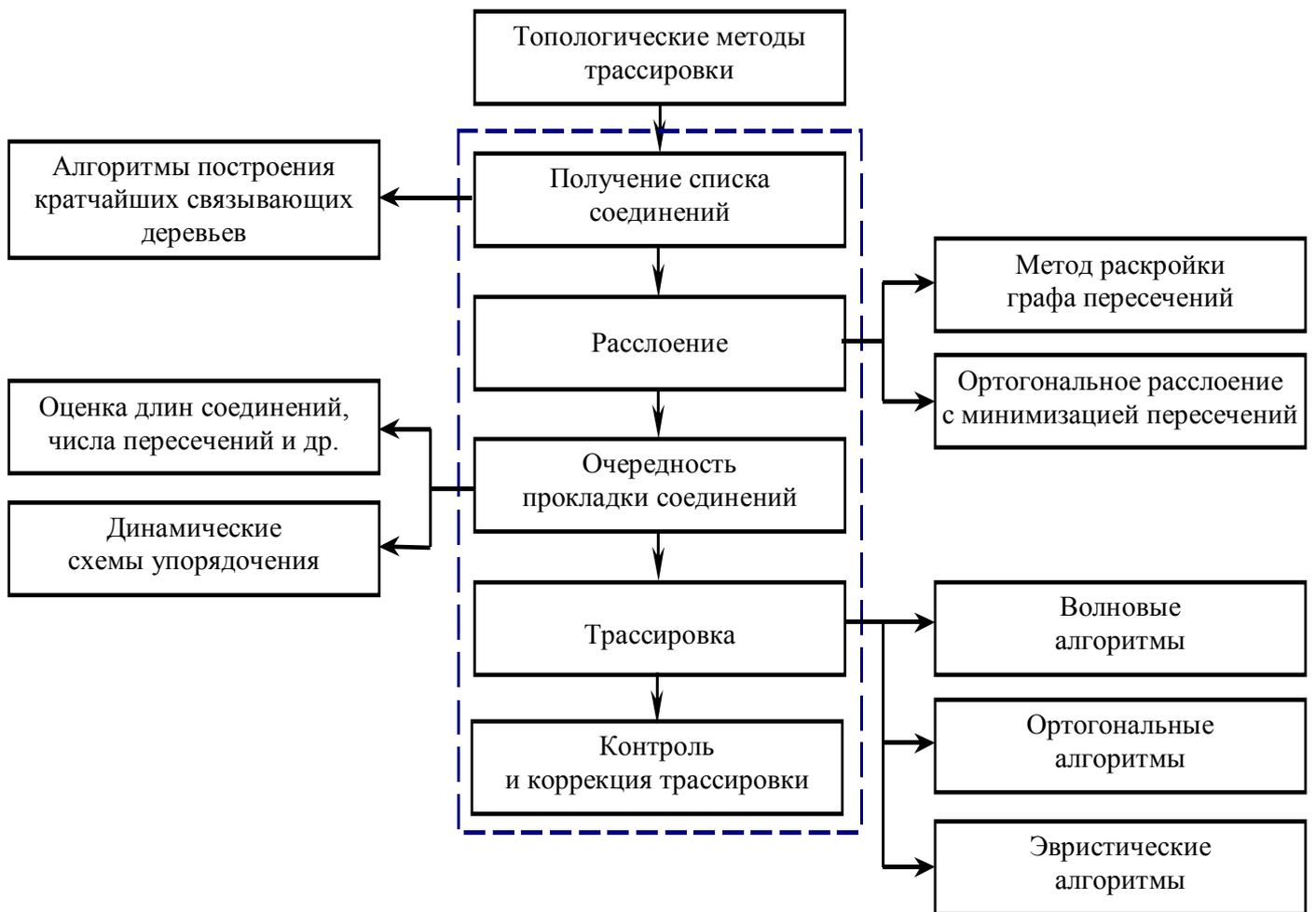


Рисунок 6.1 – Классификация алгоритмов трассировки (часть 2)

Исходная информация для решения задач трассировки соединений:

- список цепей,
- параметры конструктивных элементов
- параметры монтажного поля
- данные по размещению конструктивных элементов
- координаты выводов элемента

Задача трассировки имеет *метрический* и *топологический* аспекты.

*Метрический* аспект предполагает учет конструктивных размеров элемента и соединений контактных площадок.

*Топологический* аспект связан с выбором допустимого пространственного расположения отдельных монтажных соединений на плате при ограничениях на число пересечений и число слоев монтажной платы.

Различают методы трассировки проводных и печатных соединений (рисунок 6.1).

*Трассировка проводных соединений* более проста, т.к. цепи электрически изолированы друг от друга. Глобальная оптимизация обеспечивается локальной оптимизацией отдельной цепи.

**Трассировка печатных соединений** сложна из-за согласования метрических и топологических параметров схемы соединений и монтажного пространства.

**Требования к трассировке соединений:**

1. Соединения должны соответствовать принципиальной схеме и быть кратчайшими;
2. Число пересечений трасс в монтажном поле должно быть минимальным для МПП, либо не допускается – для ОПП
3. Распределение цепей в монтажном поле должно приближаться к равномерному;
4. Минимум числа непроведенных соединений;
5. Минимальная протяженность параллельных участков соседних проводников;
6. минимум числа изгибов проводников;
7. минимум числа слоев металлизации и числа переходов из слоя в слой (применяется при проектировании многослойных печатных плат).

Так как все перечисленные критерии являются противоречивыми, то при выборе обобщенного показателя качества указывают важность или очередность учета отдельных критериев оптимизации.

## 6.2 Трассировки проводных соединений

Несмотря на быстрое развитие печатного монтажа, удельный вес проводных соединений в ЭА остается еще достаточно большим. Как правило, проводной монтаж используют при электрическом объединении модулей третьего (субблоков, кассет) и четвертого (блоков, рам) уровней иерархии.

Трассировку проводных соединений производят двумя способами:

- 1) по прямым, соединяющим отдельные выходы модулей (монтаж внавал);
- 2) с помощью жгутов.

1) -

Трассировка заканчивается построением кратчайших деревьев (этап 1, см. рисунок 6.1, Ч1).

**Д.:**

- простота выполнения
- высокая помехоустойчивость
- позволяет до минимума сократить общую длину проводников и протяженность участков их параллельного прохождения,
- уровень паразитных наводок и время задержки сигнала в электрических соединениях относительно невелики.

**Н.:**

- высока вероятность появления в процессе монтажа ошибок (неправиль-

ное подсоединение отдельных выводов, пропуск некоторых соединений и т. п.),

☑ контроль правильности трассировки сложен.

☑ при высокой плотности монтажа обладает малой ремонтпригодностью.

2) - предполагает объединение отдельных проводников в жгуты, которые укладываются в специальные каналы, предусмотренные в монтажном поле.

При трассировке проводных соединений данным методом решаются следующие задачи (рисунок 6.1, Ч1):

1. Получение списка соединений
2. Построение кратчайших связывающих деревьев (сетей)
3. Выполнение трассировки проводов в каналах

Д.:

☑ более технологичен, так как позволяет разделить операции вязки и распайки жгутов,

☑ проще процесс контроля и устранения ошибок, допущенных при монтаже.

Н.:

☑ практически неприемлем для создания высокочастотной и чувствительной к электрическим помехам аппаратуры.

Т.О. задача трассировки проводных соединений формулируется следующим образом:

В некоторой системе координат  $XYZ$ , связанной с коммутационным пространством модуля, задано местоположение множества выводов

$$M = \{m_1, m_2, \dots, m_n\}.$$

В соответствии с электрической схемой соединений разобьем множество  $M$  на непересекающиеся подмножества  $M^{(1)}, M^{(2)}, \dots, M^{(P)}$ , каждое из которых включает в себя выводы, подлежащие электрическому объединению.

Для каждого подмножества требуется определить последовательность соединения выводов и конфигурацию проводников, обеспечивающих при заданных ограничениях минимальную суммарную длину соединений.

Необходимо учитывать также назначение цепей.

Среди различных вариантов соединения вершин (выводов) подмножества  $M^{(i)}$  целесообразно рассматривать лишь древообразные структуры (если нет – дублирование проводов).

### **Алгоритмы построения оптимальных связывающих сетей**

Все известные алгоритмы построения кратчайших связывающих сетей (КСС) основаны на последовательном выборе самых коротких связей, не образующих циклов с ранее отобранными. Процесс продолжается до тех пор, пока не будет получено минимальное дерево.

## Алгоритм Краскала (Вайнберга – Лобермана)

Пусть в некоторой системе координат  $XYZ$  задано местоположение множества точек

$$M = \{m_1, m_2, \dots, m_n\}.$$

1. Строим на множестве  $M$  полный граф  $G_n(M, U)$ .
2. Вычисляем длину всех ребер графа  $G_n(M, U)$  и
3. упорядочиваем список ребер с точки зрения их длины так, чтобы выполнялось условие

$$\forall u_i \in U [d(u_i) \leq d(u_{i+1})],$$
$$1 \leq i < r = \frac{n(n-1)}{2},$$

где  $d(u_i)$  — длина ребра  $u_i$ .

Это условие построения кортежа с возрастанием каждого очередного ребра.

Для построения дерева необходимо выбрать  $n-1$  ребер из кортежа, которые не образуют циклов.

Существуют две возможные процедуры, приводящие к построению одного и того же минимального дерева.

### Процедура 1.

*На каждом шаге просматривают список ребер (начиная с ребра, следующего за вошедшем в решение на предыдущем шаге) и к строящемуся поддереву присоединяют то ребро, которое не образует цикла с ребрами, уже включенными в решение.*

Данная процедура допускает возможность параллельного построения нескольких поддеревьев, которые затем объединяют в единую связную сеть кратчайшими ребрами.

**Н.:**

необходимость наблюдения за различными компонентами связности и проверки при выборе каждого очередного ребра условия необразования цикла для всех параллельно строящихся поддеревьев.

Эти недостатки частично устраняются во второй процедуре, которая предусматривает построение только одного поддерева.

### Процедура 2.

*На каждом шаге просматривают список ребер (начиная с первого) и к строящемуся поддереву присоединяют то ребро, которое:*

*а) еще не включено в решение;*

*б) присоединяет к поддереву новую вершину (один из концов ребра должен принадлежать вершине поддерева, другой — изолированной вершине).*

**Н.:**

необходимость на каждом шаге алгоритма начинать просмотр списка с

первого ребра, причем значительная часть просматриваемых при этом ребер может не удовлетворять условиям включения их в строящееся поддерево. Это приводит к увеличению времени решения задачи.

Указанный недостаток в значительной мере устраняется в алгоритме, предложенном Р. К. Примом.

### Алгоритм Прима

Позволяет организовать просмотр только тех ребер графа  $G_n(M, U)$ , которые связывают вершины строящегося поддерева с новыми, еще не присоединенными вершинами.

Основные принципы построения КСС при наличии дополнительного ограничения на локальные степени вершин связывающей сети:

$$\forall m_i \in M [\rho(m_i) \leq k]$$

Шаги алгоритма:

1) любая произвольная вершина  $m_i \in M$  соединяется с ближайшей соседней, образуя исходное поддерево.

Для определенности построения КСС можно начинать с ребра, инцидентного вершине  $m^1$ .

На каждом последующем шаге к строящемуся поддереву присоединяют очередное ребро минимально возможной длины, связывающее новую, еще не присоединенную вершину  $m_j$  с одной из вершин поддерева  $m_i \in M^*$ , локальная степень которой  $\rho(m_i) < k$ , т. е.

$$d(m_i, m_j) = \min_{m_f \in M^*, m_g \in M \setminus M^*} d(m_f, m_g)$$

где  $d(m_f, m_g)$  - длина ребра, соединяющего вершины

Для реализации этого алгоритма первоначально составляем матрицу длин

$$D = \left\| d_{ij} \right\|_{N \times N}$$

общий элемент которой  $d_{ij}$  равен расстоянию между  $i$ -й и  $j$ -й точками ( $N$  - число объединяемых вершин).

Просматриваем элементы первой строки матрицы  $D$  и находим минимальный из них. Пусть таким элементом оказался элемент  $g$ -го столбца, тогда весь первый и  $g$ -й столбцы матрицы  $D$  исключаем из рассмотрения, а первое соединение проводим между точками  $m_1$  и  $m_g$ .

Просматриваем первую и  $g$ -ю строки матрицы с оставшимися элементами.

Из элементов этих строк находим минимальный. Предположим, что им оказался элемент, принадлежащий  $j$ -му столбцу. Если этот элемент находится на пересечении с первой строкой, то точку  $m_j$  соединяем с  $m_1$ , если же он находится на пересечении с  $g$ -й строкой, то точку  $m_j$  соединяем с  $m_g$ , после чего из матрицы  $D$  исключаем все элементы  $j$ -го столбца.

Просматриваем первую,  $g$ -ю и  $j$ -ю строки и т.д.

Выполнение ограничения на локальную степень вершин обеспечивается проверкой в каждой просматриваемой  $i$ -й строке числа уже выбранных для построения КСС элементов  $K(i)$ . При  $K(i) = k$  все оставшиеся элементы  $i$ -й строки исключаются из рассмотрения.

**Пример.**

На плоскости в декартовой системе координат задано местоположение девяти точек (рисунок 6.2). Расстояние между любыми двумя точками  $m_i$  и  $m_j$  равно

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

Требуется для заданного множества точек  $M$  определить минимальное связывающее дерево при условии:

$$\forall m_i \in M [\rho(m_i) \leq 3]$$

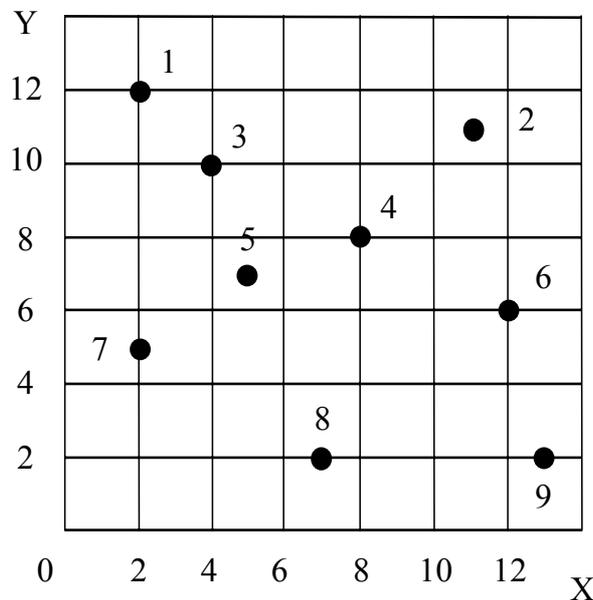


Рисунок 6.2 – Пример

**Решение.**

Составляем матрицу длин:

$$D = \begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 0 & 10 & \langle 4 \rangle & 10 & 8 & 16 & 7 & 15 & 21 \\ 2 & 10 & 0 & 8 & 6 & 10 & \langle 6 \rangle & 15 & 13 & 11 \\ 3 & 4 & 8 & 0 & 6 & \langle 4 \rangle & 12 & 7 & 11 & 17 \\ 4 & 10 & \langle 6 \rangle & 6 & 0 & 4 & 6 & 9 & 7 & 11 \\ 5 & 8 & 10 & 4 & \langle 4 \rangle & 0 & 8 & \langle 5 \rangle & 7 & 13 \\ 6 & 16 & 6 & 12 & 6 & 8 & 0 & 11 & 9 & \langle 5 \rangle \\ 7 & 7 & 15 & 7 & 9 & 5 & 11 & 0 & 8 & 14 \\ 8 & 15 & 13 & 11 & 7 & 7 & 9 & 8 & 0 & 6 \\ 9 & 21 & 11 & 17 & 13 & 13 & 5 & 14 & \langle 6 \rangle & 0 \end{array}$$

1) Просматриваем первую строку матрицы и выбираем элемент  $d_{13}$ , являющийся минимальным в этой строке (при наличии нескольких элементов с одинаковыми минимальными значениями выбираем элемент с меньшим индексом  $j$ ).

2) Помечаем элемент  $d_{13}$ ;  $K(1) = K(3) = 1$ . Исключаем из рассмотрения (вычеркиваем) все элементы первого и третьего столбцов.

3) Просматриваем первую и третью строки. Выбираем элемент  $d_{35}$ ;  $K[3] = 2$ ,  $K[5] = 1$ . Исключаем из рассмотрения элементы пятого столбца.

4) Просматриваем первую, третью и пятую строки. Выбираем элемент  $d_{54}$ ;  $K(5) = 2$ ,  $K(4) = 1$ . Исключаем из рассмотрения элементы четвертого столбца.

5) Просматриваем первую, третью, четвертую и пятую строки. Выбираем элемент  $d_{57}$ ;  $K(5) = 3$ ,  $K(7) = 1$ . Так как  $K[5] = k$ , то исключаем из рассмотрения элементы седьмого столбца и пятой строки.

6) Продолжая процесс построения КСС аналогичным образом, выбираем элементы  $d_{42}$ ,  $d_{26}$ ,  $d_{69}$ ,  $d_{98}$ .

Минимальное связывающее дерево, полученное в результате решения, приведено на рисунке 6.3. Суммарная длина его ребер равна  $D = 40$ .

Если локальная степень вершин не должна превышать двух, то в результате решения будут выбраны элементы  $d_{13}$ ,  $d_{35}$ ,  $d_{54}$ ,  $d_{42}$ ,  $d_{26}$ ,  $d_{69}$ ,  $d_{98}$ ,  $d_{17}$ . Суммарная длина ребер при этом равна 42.

Алгоритм Прима находит широкое применение в системах автоматизированного проектирования проводных соединений ЭА.

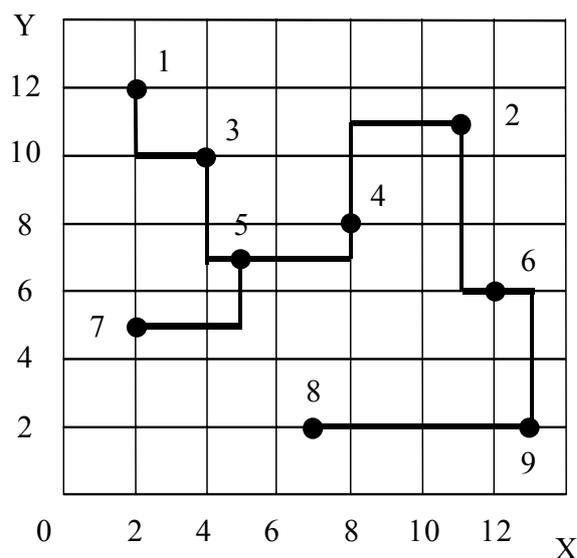


Рисунок 6.3 – Решение

### Алгоритмы трассировки проводов в каналах

При выполнении электрических соединений модулей третьего и четвертого уровней иерархии обычно используют жгутовой монтаж, при котором проводники укладывают в нормализованные каналы, расположенные в монтажном пространстве во взаимно перпендикулярных направлениях.

Такую канальную конструкцию можно представить в виде ортогональной несимметрической сети  $G(A, B)$  со множеством узлов  $A$  и множеством дуг  $B$ .

Термин «сеть» используется для того, чтобы подчеркнуть характерную особенность рассматриваемой задачи, связанную с приписанием каждой дуге  $b_{ij}$ , соединяющей узлы  $a_i, a_j$  некоторого положительного числа  $c_{ij}$ , определяющего максимальное число проводников, которое можно укладывать в соответствующем канале, интерпретируемом дугой  $b_{ij}$ . Величину  $c_{ij}$  называют **пропускной способностью дуги**  $b_{ij}$ .

При любом допустимом варианте трассировки

$$\forall b_{ij} \in B \left[ 0 \leq x_{ij} \leq c_{ij} \right], \quad (6.1)$$

где  $x_{ij}$  - дуговой поток или поток по дуге  $b_{ij}$ , равный числу проводников в канале, соединяющем точки  $i$  и  $j$ .

Разобьем множество  $A$  на два непересекающихся подмножества:

$A_1$  - подмножество узлов, соответствующих электрическим контактам модулей и разъемов схемы;

$A_2$  - подмножество узлов, в которых допускается изменение направления прокладывания проводников (точки пересечения вертикальных и горизонтальных каналов).

Подмножество  $A_l$  в свою очередь состоит из двух непересекающихся подмножеств:

$A_s$  - источники и  $a_t$  - стоки, определяющие, соответственно, электрические контакты модулей и разъемов.

Для любого узла

$$a_j \in A = A_s \cup A_T \cup A_2$$

сети  $G(A, B)$  должно выполняться условие

$$\sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -X_j, & \text{если } a_j \in A_s \\ 0, & \text{если } a_j \in A_2 \\ X_j, & \text{если } a_j \in A_T \end{cases}, \quad (6.2)$$

где  $X_j \geq 0$ .

Первая сумма берется по дугам, ведущим в узел  $a_j$ , а вторая - по дугам, ведущим из этого узла.

Положительное число  $X_j$  определяет величину потока, входящего (выходящего) в  $j$ -й узел.

Таким образом, в каждый узел  $a_{ij}$ , за исключением источников и стоков, входит такое же количество потока, какое из него выходит (условие сохранения потока (=0)).

Полный поток из  $A_s$  в  $A_T$

$$X = \sum_{a_j \in A_T} X_j = - \sum_{a_j \in A_s} X_j. \quad (6.3)$$

В реальных конструкциях пропускные способности каналов ограничены, поэтому естественна постановка задачи отыскания максимального потока в сети, которая сводится к нахождению такого распределения потоков (проводников) по отдельным каналам, которое максимизирует целевую функцию (6.3) при ограничениях (6.1) и (6.2).

Решаются данные задачи методами линейного целочисленного программирования либо на методе расстановки пометок.

## 7 алгоритмы и модели трассировки печатных соединений в эа

Проектирование печатного монтажа является одной из самых сложных задач автоматизации проектирования ЭА. Для ее решения предложено большое число различных алгоритмов.

### 7.1 Постановка задачи трассировки печатных соединений

Задача проектирования печатного монтажа может быть сформулирована следующим образом.

На коммутационной поверхности заданы своими координатами  $(x, y)$  множество конструктивных элементов  $R = \{r_1, r_2, \dots, r_n\}$ . Выводы этих элементов образуют некоторое множество из  $L$  связных подмножеств:  $\varepsilon = \{C_1, C_2, \dots, C_L\}$ , где каждое  $l$ -е подмножество  $C_l$  объединяет  $n_l$  выводов конструктивных элементов из множества  $R$  в соответствии с принципиальной электрической схемой.

Кроме того, заданы расположение групп контактных площадок разъемов и монтажных отверстий, а также ряд требований, предъявляемых к топологии платы:

- минимальная ширина проводников и зазора между ними,
- размеры контактных площадок,
- число слоев металлизации и способы перехода с одного слоя на другой

и т. п.

Требуется с учетом заданных конструкторско-технологических ограничений соединить выводы конструктивных элементов внутри каждого подмножества  $C_l \subset \varepsilon$  так, чтобы полученные соединения отвечали выбранному показателю качества.

В общем виде *задачу трассировки печатных соединений можно свести* к

- 1) построению бесперекрестного минимального леса и
- 2) отысканию кратчайшего пути между его вершинами (трассировка соединений).

Обе эти задачи можно рассматривать как независимые, алгоритмически разрешимые.

1) Построение *бесперекрестного минимального леса* сводится к поиску таких связывающих деревьев  $G_l$ , для каждого из подмножеств  $C_l$ , которые минимизируют целевую функцию

$$F = \sum_{l=1}^L \sum_{n=1}^{N_l-1} (d_{ij}^n)_l$$

где  $(d_{ij}^n)_l$  - длина  $n$ -й связи в выбранной метрике между точками  $i$  и  $j$  (контактами выводов), принадлежащими подмножеству  $S_l$ , объединяющему  $N_l$  контактов.

Применение алгоритмов Краскала (Вайнберга-Лобермана) и Прима позволяет найти минимальное связывающее дерево лишь относительно ранее проведенных цепей, поэтому при трассировке печатных соединений практически не применяются.

Единственно достоверным способом получения оптимального результата в этих условиях является полный перебор вариантов последовательностей построения бесперекрестного леса, что на практике невозможно.

Характерной особенностью печатного монтажа является возможность кроме соединений типа «вывод-вывод», применяемых в проводном монтаже, использовать соединения типа «вывод-проводник» и «проводник-проводник».

Оптимизация подобных цепей предполагает построение для каждого множества объединяемых вершин  $M^i = \{m_1^{(i)}, m_2^{(i)}, \dots, m_n^{(i)}\}$  минимальных деревьев, допускающих введение произвольного числа дополнительных вершин, образующих множество  $D = \{d1, d2, \dots, dk\}$ . (т.е. точки соединения и разветвления печатных проводников)

Практически все методы построения минимальных связывающих деревьев не учитывают ограничения на размеры монтажного поля, толщину печатных проводников и величину зазора между ними. В результате значительную часть найденных деревьев оказывается невозможным реализовать в виде электрических цепей печатной платы.

Указанный недостаток, а также большое время, затрачиваемое на построение минимальных связывающих деревьев, привели к тому, что данные алгоритмы применяют лишь для отыскания пар контактов, подлежащих соединению.

2) *Трассировку соединений* осуществляют с помощью алгоритмов, основанных на методах динамического программирования.

Общим для этих алгоритмов является разбиение монтажного поля на ячейки, размер и форма которых определяют плотность и конфигурацию печатных проводников (равносторонние треугольники, квадраты, шестиугольники).

Наибольшее распространение на практике получило разбиение рабочего поля на правильные квадраты, что обеспечивает простую адресацию ячеек в прямоугольной системе координат и привычную форму соединений.

Покрытие поля равносторонними треугольниками является наименее удачным, так как при реализации алгоритма трассировки потребуются введение дополнительной операции сглаживания углов. Кроме того, использование треугольных и шестиугольных ячеек усложняет их адресацию.

Размеры ячеек определяются конструктивно-технологическими требованиями, предъявляемыми к печатному монтажу. Так как в каждой ячейке обычно размещается только один вывод или печатный проводник, максимальные

размеры ячеек определяются допустимой точностью воспроизведения проводников.

Минимальные размеры ячеек обуславливаются возможностью ОЗУ ЭВМ и соотношением

$$d \geq b_n - l$$

где  $d$  - расстояние между центрами соседних ячеек;  $b_n$  - минимальная ширина печатного проводника;  $l$  - минимальное расстояние между соседними проводниками.

Соединение выводов конструктивных элементов осуществляется в результате последовательного заполнения ячеек трассами, конфигурация которых является локально оптимальной (соединения проводятся оптимальным образом лишь по отношению к ранее построенным проводникам) в соответствии с выбранными критериями трассировки.

При последовательном процессе проведения трасс (поскольку многие соединения конкурируют между собой) число разведенных цепей и их конфигурация определяются последовательностью трассировки отдельных соединений.

Известные алгоритмы трассировки печатных плат можно условно разбить на три большие группы:

**1) волновые алгоритмы,**

Основаны на идеях Ли. Получили широкое распространение в существующих САПР,

**Д.:**

✓ позволяют легко учитывать технологическую специфику печатного монтажа со всей совокупностью конструктивных ограничений.

✓ всегда гарантируют построение трассы, если путь для нее существует;

**2) ортогональные алгоритмы,**

**Д.:**

✓ обладают большим быстродействием, чем алгоритмы 1)

✓ реализация их на ЭВМ требует в 75—100 раз меньше вычислений по сравнению с волновыми алгоритмами.

**Н.:**

✓ получение большого числа переходов со слоя на слой,

✓ отсутствие 100%-ной гарантии проведения ряда трасс,

✓ большое число параллельно идущих проводников.

Такие алгоритмы применяют при проектировании печатных плат со сквозными металлизированными отверстиями.

**3) алгоритмы эвристического типа,**

В настоящее время широко распространены (критичны к объему памяти и быстродействию ЭВМ). Частично основаны на эвристическом приеме поиска пути в лабиринте. При этом каждое соединение - проводится по кратчайшему пути, обходя встречающиеся на пути препятствия.

## 7.2 Волновые алгоритмы. Волновой алгоритм Ли

Классический пример использования методов динамического программирования для решения задач трассировки печатных соединений.

**Основные принципы построения трасс** с помощью волнового алгоритма:

Все ячейки монтажного поля подразделяют на **занятые и свободные**.

**Занятыми** считаются ячейки, в которых уже расположены проводники, построенные на предыдущих шагах, или находятся монтажные выводы элементов, а также ячейки, соответствующие границе платы и запрещенным для прокладки участков.

Каждый раз при проведении новой трассы можно использовать лишь свободные ячейки, число которых по мере проведения трасс сокращается.

На множестве свободных ячеек коммутационного поля моделируют волну влияния из одной ячейки в другую, соединяемых впоследствии общим проводником.

Первую ячейку, в которой зарождается волна влияний, называют **источником**, а вторую - **приемником** волны.

Чтобы иметь возможность следить за прохождением фронта волны влияний, его фрагментам на каждом этапе присваивают некоторые **веса**:

$$P_k = P_{k-1} + \psi(f_1, f_2, \dots, f_g)$$

где  $P_k$  и  $P_{k-1}$  - веса ячеек  $k$ -го и  $(k-1)$ -го фронтов;  $\psi(f_1, f_2, \dots, f_g)$  - весовая функция, являющаяся показателем качества проведения пути, каждый параметр которой  $f_i (i = 1, 2, \dots, g)$  характеризует путь с точки зрения одного из критериев качества (длины пути, числа пересечений и т. п.).

На  $P_k$  накладывают одно ограничение - веса ячеек предыдущих фронтов не должны быть больше весов ячеек последующих фронтов.

Фронт распространяется только на соседние ячейки, которые имеют с ячейками предыдущего фронта либо общую сторону, либо хотя бы одну общую точку. Процесс распространения волны продолжается до тех пор, пока расширяющийся фронт не достигнет приемника или на  $\xi$ -ом шаге не найдется ни одной свободной ячейки, которая могла бы быть включена в очередной фронт, что соответствует случаю невозможности проведения трассы при заданных ограничениях.

Если в результате распространения волна достигла приемника, то осуществляют «**проведение пути**», которое заключается в движении от приемника к источнику по пройденным на этапе распространения волны ячейкам, следя за тем, чтобы значения  $P_k$  монотонно убывали. В результате получают путь, соединяющий эти две точки.

Из описания алгоритма следует, что все условия, необходимые для проведения пути, закладываются в правила приписания веса ячейкам.

Чтобы исключить неопределенность при проведении пути для случая, когда несколько ячеек имеют одинаковый минимальный вес, вводят понятие путевых координат, задающих предпочтительность проведения трассы. Каждое направление кодируют двоичным числом по  $\text{mod } q$ , где  $q$  - число просматриваемых соседних ячеек. При этом чем более предпочтительно то или иное направление, тем меньший числовой код оно имеет.

Например, если задаться приоритетным порядком проведения пути сверху, справа, снизу и слева, то коды соответствующих путевых координат будут 00, 01, 10 и 11.

Приписание путевых координат производят на этапе распространения волны. При проведении пути движение от ячейки к ячейке осуществляют по путевым координатам.

### Примеры

(при использовании различных критериев качества)

Во всех примерах задан приоритетный порядок проведения пути: сверху, справа, снизу и слева:

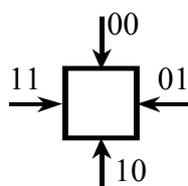


Рисунок 7.1

Вместо числовых кодов для обозначения соответствующих путевых координат будем использовать стрелки, указывающие их направление

### Проведение пути минимальной длины

Задано множество ячеек коммутационного поля, на котором построено некоторое число проводников (рисунок 7.2). Построить новый проводник между точками X и Y так, чтобы он не пересекал ранее построенные проводники и имел минимально возможную длину.

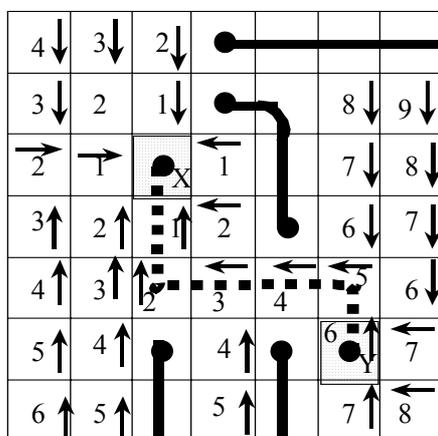


Рисунок 7.2

Из точки  $X$ , вес которой положим равным нулю, строим расширяющийся фронт волны влияний, распространяющийся на все свободные и соседние с этой точкой ячейки.

Каждая ячейка первого фронта генерирует следующий фронт волны, который занимает все свободные и соседние с первым фронтом ячейки и т. д.

Вес ячейки  $k$ -го фронта считаем равным весу соседней ячейки  $(k-1)$ -го фронта плюс единица:

$$P_k = P_{k-1} + 1$$

Процесс распространения волны продолжаем до тех пор, пока не достигнем ячейки с точкой  $Y$ .

Процесс поиска пути из  $X$  в  $Y$  в соответствии с рассмотренным алгоритмом для данного варианта трассировки показан на рисунке 7.2. В каждой ячейке указаны приписанные ей на этапе распространения волны путевые координаты и веса. Ячейка и достигается при построении 6-го фронта волны.

Проведение пути начинаем с ячейки  $Y$ .

Просматриваем окрестность точки приемника и находим ячейку, которая в наиболее предпочтительном направлении имеет вес на единицу меньше. Перемещаемся в эту ячейку и отмечаем след перехода. Процесс продолжаем до тех пор, пока след не приведет в точку  $X$ . На рисунке 7.2 вновь проведенный путь изображен пунктиром.

В данном случае путевая координата несет избыточную информацию. Действительно, при проведении пути использовались только данные о весе просматриваемой ячейки, а приоритет его проведения учитывался последовательностью просмотра этих ячеек.

### **Проведение пути с минимальным числом пересечений**

При такой постановке задачи *занятыми* считают ячейки, в которых:

- а) находятся выводы конструктивных элементов,
- б) имеются изгибы или пересечения ранее построенных проводников,
- в) ячейки, в которых направление проводников совпадает с путевой координатой строящегося пути.

Такое определение занятости допускает пересечение проводников во взаимно перпендикулярных направлениях и запрещает их параллельное прохождение в одной ячейке.

Вес незанятой ячейки  $k$ -го фронта считаем равным весу соседней ячейки  $(k-1)$ -го фронта, если в этой ячейке нет ранее построенного проводника и на единицу больше в противном случае.

При этом в  $k$ -й фронт включаем лишь те ячейки, которые имеют минимальный вес. Это условие не позволяет расширяющемуся фронту волны пересечь построенные проводники до тех пор, пока окрестность ячейки-источника не заполнит замкнутую область, ограниченную проводниками и занятыми ячейками. Если при этом ячейка-приемник достигнута не будет, то фронт сразу пересекает все проводники, образующие границу области.

При втором и последующих пересечениях процесс повторяется.

После достижения ячейки-приемника проводник проводится согласно путевым координатам, присвоенным ячейкам в процессе распространения волны. На рисунке 7.3 показан вариант проведения пути между точками X и Y для рассматриваемого случая. Вес, присваиваемый ячейке Y, равен числу пересечений строящегося проводника с ранее проведенными.

В этом алгоритме избыточной информацией, которая не учитывается при проведении пути, является вес ячейки. От его присвоения можно отказаться, если последовательно, учитывая приоритет направления, присваивать путевые координаты всем ячейкам каждой очередной области, ограниченной проводниками, переходя от одной области к другой по мере их заполнения.

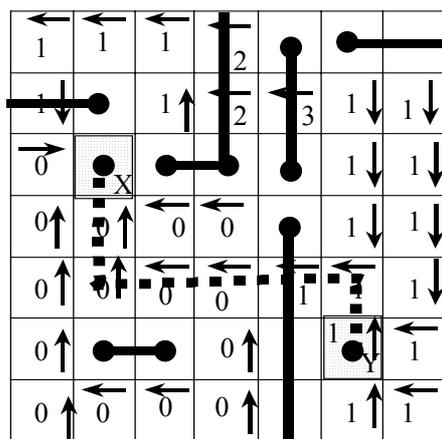


Рисунок 7.3

### Параллельная оптимизация пути по нескольким параметрам

Возможна.

Практически в любой задаче трассировки печатных проводников приходится учитывать не один, а несколько критериев качества. Если удастся задать количественную оценку важности отдельных критериев, то задача может быть решена с помощью алгоритма Ли.

В качестве примера рассмотрим один из возможных вариантов присвоения веса незанятой ячейке  $k$ -го фронта:

- 1)  $P_k = P_{k-1} + 1$  если в данной и соседних ячейках нет ранее построенных проводников и путевая координата не меняет своего направления;
- 2)  $P_k = P_{k-1} + 2$ , если в соседних ячейках нет ранее построенных проводников, но путевая координата меняет свое направление;
- 3)  $P_k = P_{k-1} + 3$ , если в данной ячейке путевая координата не меняет своего направления и нет ранее построенного проводника, но в соседней ячейке такой проводник есть;
- 4)  $P_k = P_{k-1} + 5$ , если в данной ячейке происходит пересечение с ранее построенным проводником.

**!!! Для платы (рисунок 7.3) построить связь между X и Y с учетом данных ограничений.**

### **Недостатки волнового алгоритма Ли:**

1. малое быстродействие;
2. большой объем оперативной памяти ЭВМ, необходимый для хранения информации о текущем состоянии всех ячеек коммутационного поля;
3. возможность построения лишь соединений типа «вывод - вывод».

### **7.3 Модификации волнового алгоритма**

1. Метод встречной волны.
2. Метод соединения комплексами.
3. Лучевой алгоритм трассировки.

#### **Метод встречной волны**

В данном методе источниками волн являются обе ячейки, подлежащие электрическому объединению. При этом на каждом  $k$ -ом шаге поочередно строят соответствующие фронты первой и второй волн, распространяющихся из этих ячеек. Процесс продолжается до тех пор, пока какая-либо ячейка из фронта первой волны не попадет во фронт второй волны или наоборот. Проведение пути осуществляют изданной ячейки в направлении обоих источников по правилам, описанным в волновом алгоритме Ли.

**Д.:**

1. время, затрачиваемое на этапе распространения волны, уменьшаются примерно вдвое.

**Н.:**

1. необходимость выделения дополнительного разряда памяти на каждую рабочую ячейку поля для хранения информации о принадлежности ее к первой или второй волне.
2. возможность построения лишь соединений типа «вывод - вывод».

#### **Метод соединения комплексами:**

Пусть необходимо объединить некоторое множество точек  $M = \{m_1, m_2, \dots, m_n\}$ .

1. Выбираем в качестве источника произвольную точку

$$m_i \in M$$

и распространяем из нее волну влияний до встречи с какой-либо точкой

$$m_j \in M$$

2. Проводим путь, соединяющий точки  $m_i$  и  $m_j$ .
3. Вновь перейдем к процессу «распространения волны», выбрав в качестве источника только что построенный проводник, и заканчиваем его при встрече с какой-либо точкой

$$m_k \in M \setminus \{m_i, m_j\}$$

4. Проведя путь, получаем соединение уже трех точек множества  $M$ .

5. Теперь источником волны является дерево, состоящее из построенных проводников, и т. д.

**Д.:**

1. возможность присоединения каждой очередной точки (начиная с третьей), к любой точке ранее построенных соединений,

2. сокращение общей длины печатных проводников

3. увеличение числа разводимых цепей

4. возможность построения соединений типа «вывод - проводник» и «проводник - проводник».

Быстродействие алгоритма можно повысить, применив метод встречной волны, если это позволяет объем памяти ЭВМ. В данном случае источниками «распространения волн» являются все ранее построенные проводники и ячейки, подлежащие объединению.

Для дополнительного сокращения затрат машинного времени производят неполную очистку ячеек рабочего поля после подсоединения каждой очередной точки к строящейся цепи. Для этого достаточно запомнить в специальном счетчике радиус  $R$  окрестностей, построенных до встречи на предыдущем шаге, и перед следующим шагом очистить окрестности радиусом  $R$  только тех двух источников, которые только что были соединены. Для вновь проведенного соединения строят окрестность радиусом  $R$ . Если не произошло встречи с какой-либо уже построенной окрестностью, дальнейший процесс «распространения волны» продолжается для всех источников, если встреча состоялась, то вновь переходят к проведению пути и т. д.

### **Лучевой алгоритм трассировки**

В данном алгоритме выбор ячеек для определения пути между соединяемыми точками А и В производят по заранее заданным направлениям, подобным лучам.

**Д.:**

1. Это позволяет сократить число просматриваемых алгоритмом ячеек, а следовательно, и время на анализ и кодировку их состояния,

**Н.:**

1. приводит к снижению вероятности нахождения пути сложной конфигурации,

2. усложняет учет конструктивных требований к технологии печатной платы.

Работа алгоритма заключается в следующем.

Задается число лучей, распространяемых из точек А и В, а также порядок присвоения путевых координат (обычно число лучей для каждой ячейки-

источника принимается одинаковым).

Лучи  $A^{(1)}, A^{(2)}, \dots, A^{(n)}$  и  $B^{(1)}, B^{(2)}, \dots, B^{(n)}$  считают одноименными, если они распространяются из одноименных источников  $A$  или  $B$ .

Лучи  $A^{(i)}$  и  $B^{(i)}$  являются разноименными по отношению друг к другу.

Распространение лучей производят одновременно из обоих источников до встречи двух разноименных лучей в некоторой ячейке  $C$ .

Путь проводится из ячейки  $C$  и проходит через ячейки, по которым распространялись лучи.

При распространении луча может возникнуть ситуация, когда все соседние ячейки будут заняты. В этом случае луч считается заблокированным и его распространение прекращается.

**Пример** (рисунок 7.4).

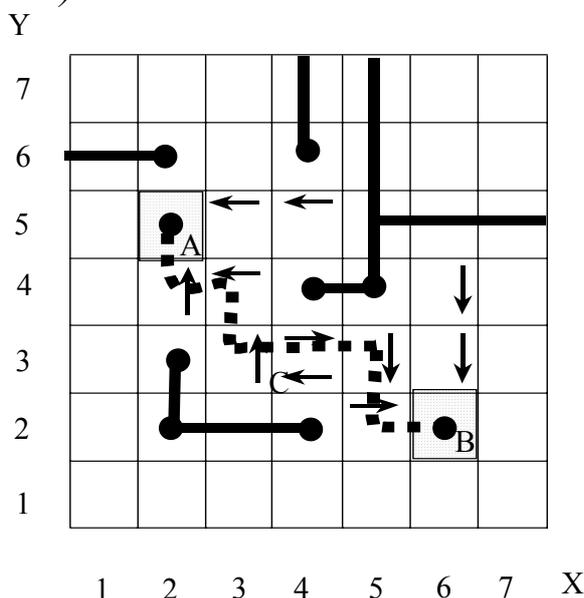


Рисунок 7.4 – Пример для лучевого алгоритма трассировки

Выбираем для источников  $A$  и  $B$  по два луча с противоположными направлениями:  $A^{(1)}$  - вверх, влево;  $A^{(2)}$  - влево, вверх;  $B^{(1)}$  - вниз, вправо;  $B^{(2)}$  - вправо, вниз.

Если ячейка  $B$  располагалась бы не справа, а слева от  $A$ , то путевые координаты влево и вправо необходимо было бы поменять.

На первом шаге просматривают ячейки с координатами (2,4), (5,2) и (6,3).

На втором шаге луч  $B^{(1)}$  и луч  $A^{(2)}$  оказываются заблокированными.

Лучи  $B^{(2)}$  и  $A^{(1)}$  встречаются в ячейке  $C$  с координатами (4,3) на четвертом шаге.

Проводим трассу.

Например, если бы ячейка с координатами (4,3) была занята (имелись проводники), то все лучи оказались бы заблокированными и решение найдено не было, хотя путь из  $A$  в  $B$  провести можно.

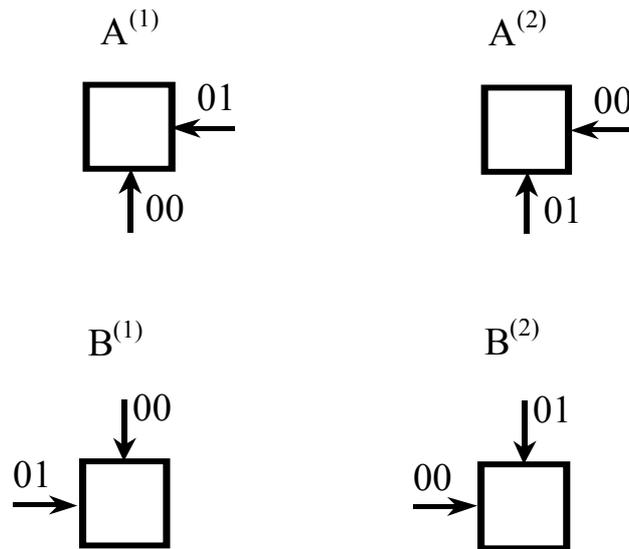


Рисунок 7.5 –Путевые координаты для лучей

Обычно с помощью лучевого алгоритма удается построить до (70-80)% трасс, остальные проводят, используя волновой алгоритм или вручную.

Его применение особенно выгодно при проектировании печатных плат с невысокой плотностью монтажа.

#### 7.4 Эвристические алгоритмы трассировки

**Д.:**

1. относятся к числу наиболее быстродействующих и простых в программировании.

**Н.:**

1. заложенный в их основу приоритетный (постоянный) порядок построения трассы и обхода препятствий влечет за собой неоптимальность получаемого результата.

Поэтому эти алгоритмы применяют в тех случаях, когда основным является скорость решения задачи, а к качеству трассировки жестких требований не предъявляется.

**Пример.**

На каждом шаге из числа свободных соседних ячеек выберем ту, в которой расстояние до ячейки-цели уменьшается (увеличивается) на максимально (минимально) возможную величину. При прочих равных условиях выбираем направление, соответствующее минимальному номеру путевой координаты. Если на пути движения встречается препятствие, то его обход осуществляем по первому свободному направлению, исследуя состояния соседних ячеек в порядке выбранного приоритета.

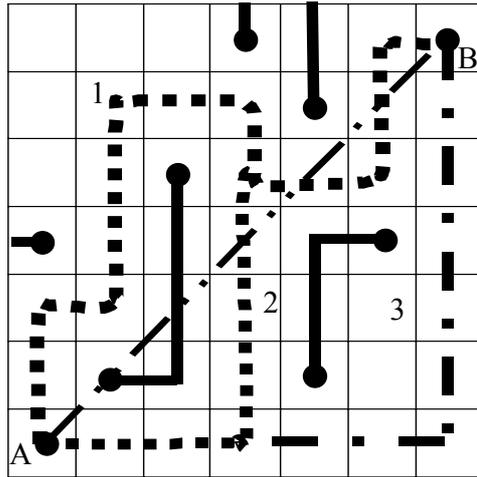


Рисунок 7.6 –Пример для эвристического алгоритма трассировки

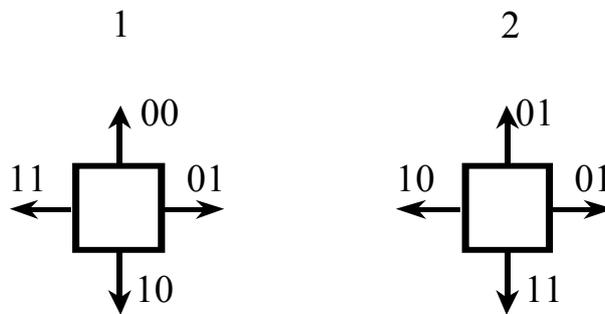


Рисунок 7.7 – Путевые координаты для лучей

Зададим следующий приоритетный порядок проведения пути: вверх, вправо, вниз, влево (вариант 1).

Общее направление движения должно происходить по ломаной линии минимальной длины или, если это возможно, по прямой, соединяющей объединяемые точки.

*Считаем, что трасса удаляется от цели, если на очередном шаге алгоритма расстояние до данной прямой возрастает.*

Пусть минимальное расстояние до прямой АВ на первом шаге алгоритма (движение начинаем из точки А) соответствует верхней и правой ячейкам. Выбираем верхнюю ячейку, так как она находится на более приоритетном направлении. Повторяем аналогичную процедуру для только что выбранной ячейки и т. д. В результате строящийся проводник обойдет препятствие слева.

При изменении порядка движения: вправо, вверх, влево, вниз - конфигурация трассы изменится и препятствие будет обойдено справа (вариант 2).

Построенные с помощью данного алгоритма трассы не являются оптимальными. Для сравнения на рисунке 7.7 штрихпунктиром (вариант 3) изображен путь, найденный с помощью волнового алгоритма при оптимизации соединения по двум критериям: *минимуму длины и числа изгибов проводника.*

нения по двум критериям: *минимуму длины и числа изгибов проводника*.

**Другой вариант эвристического алгоритма**, не требующий разбиения монтажного поля на ячейки, основан на проведении соединений по кратчайшему пути в обход препятствий, представляемых в виде прямоугольников, задаваемых координатами своих углов. Размеры прямоугольников выбирают такими, чтобы по его сторонам могли прокладываться новые проводники. При встрече с препятствием трасса проводится по периметру соответствующего прямоугольника. Каждое конкретное направление обхода оценивается с точки зрения удлинения строящегося проводника и числа возможных пересечений его с ранее построенными и еще непроведенными трассами. При этом выбирается то направление, которому соответствует лучшее значение оценки. При невозможности обхода очередного препятствия в программе предусмотрена процедура возврата к предыдущему препятствию и выбору иного варианта его обхода. Использование этой процедуры позволяет сократить число неразведанных цепей. Для экономии оперативной памяти ЭВМ в алгоритме применен принцип фрагментального проведения трасс, при котором все монтажное поле разбивается на отдельные дискретные элементы, хранящиеся во внешнем ЗУ и вызываемые в оперативную память по мере необходимости при прокладывании проводника внутри соответствующего дискрета. Это создает благоприятные условия для трассировки плат практически любого размера.

## **7.5 Особенности трассировки соединений в многослойных печатных платах**

Высокая плотность компоновки элементов современной ЭА привела к необходимости использования для их коммутации многослойных печатных плат (МПП). В зависимости от способа выполнения межслойных соединений различают МПП:

- 1) с открытыми контактными площадками;
- 2) со сквозными металлизированными отверстиями.

В первом случае переход из слоя в слой разрешается только по ножкам модулей, что делает данный способ реализации МПП достаточно технологичным. Однако, необходимость построения каждого соединения в пределах какого-либо одного слоя приводит к увеличению средней длины печатных проводников и усложнению их конфигурации, в результате чего коэффициент заполнения проводниками каждого слоя получается сравнительно низким, а число слоев платы - относительно большим. Поэтому основным критерием оптимизации данного вида МПП является минимум числа слоев платы.

Во втором случае для исключения пересечений печатных проводников разрешается трассировать отдельные соединения в разных слоях платы, используя в качестве контактных переходов сквозные металлизированные отверстия, что позволяет уменьшить число слоев металлизации по сравнению с пре-

дыдущим вариантом МПП. Однако данный способ изготовления печатных плат менее технологичен, так как требует дополнительной механической обработки (сверление отверстий) и гальванопокрытия контактных переходов. Кроме того, надежность подобных переходов ниже надежности печатных проводников, что приводит к снижению выхода годных плат, а следовательно, к росту их стоимости. Поэтому основным критерием оптимизации таких МПП является минимум числа переходов из слоя в слой.

Таким образом, каждый из рассмотренных методов изготовления МПП имеет свои достоинства и недостатки, которые необходимо учитывать для обоснованного выбора конкретного варианта печатных плат.

**МПП с открытыми контактными площадками.** Процесс проектирования МПП с открытыми контактными площадками можно условно разбить на три этапа:

- 1) построение оптимальных связывающих деревьев;
- 2) разбиение ребер минимального леса на непересекающиеся подмножества, определение очередности построения соединений каждого слоя платы;
- 3) трассировка печатных проводников.

#### **МПП со сквозными металлизированными отверстиями**

Используется ортогональный монтаж, в котором все соединения проводят параллельно координатным осям, образуя систему горизонтальных и вертикальных магистралей, которые располагаются в разных слоях. Переходы из слоя в слой осуществляются в местах пересечения магистралей.

## 8 Программное обеспечение САПР. Пакет P-CAD 2000/04

### 8.1 Общие сведения о системе проектирования P-CAD

До последнего времени наиболее популярной в России программой разработки печатных плат (ПП) являлась система **P-CAD** для **DOS** версии **4.5**, появившаяся в конце 1989 г. Для нее созданы кириллические шрифты, обширные библиотеки и драйверы для сопряжения с используемым технологическим оборудованием, разработаны подробные инструкции по применению.

Сравнительно недавно получили распространение более поздние версии **P-CAD 8.x**, в которых, по сравнению с P-CAD 4.5, изменен формат файлов баз данных, что затрудняет передачу технической документации на предприятия-изготовители ПП. Дополнительным препятствием к их применению служит отсутствие возможности непосредственного нанесения на чертежи надписей по-русски. Все версии P-CAD для DOS обладают примерно одинаковыми возможностями. При этом P-CAD 8.7 значительно отличается от P-CAD 4.5, но не так существенно, как хотелось бы.

Система P-CAD несколько раз меняла владельцев, в настоящее время она принадлежит австралийской компании Protel International ([www.protel.com](http://www.protel.com)). Ее предыдущий владелец ACCEL Technologies ([www.acceltech.com](http://www.acceltech.com)) преобразован в американский филиал этой компании, интересы которой в странах СНГ и Балтии представляет московская фирма «Родник Софт» ([www.rodnik.ru](http://www.rodnik.ru)).

Система **P-CAD** для **Windows** выпущена в феврале 1996 г. и получила название **ACCEL EDA**. После смены владельца ей в начале 2000 г. вернули старое название **P-CAD** (**ACCEL EDA 15.1 стала называться P-CAD 2000**).

#### **Возможности:**

P-CAD выполняет полный цикл проектирования ПП, включающий в себя

- графический ввод схемы,
- «упаковку» (перенос) схемы на ПП,
- ручное размещение компонентов,
- ручную, интерактивную или автоматическую трассировку проводников,
- контроль ошибок в схеме и ПП
- и выпуск конструкторской и технологической документации.

Применение сопутствующих программ позволяет выполнять *моделирование схем и анализ паразитных эффектов, присущих реальным ПП, до их изготовления*, что обеспечивает преимущества P-CAD по сравнению с другими САПР.

#### **Недостатки предыдущих версий:**

В **P-CAD 4.5** разрешающая способность редактора ПП была недостаточной: 1 мил (0,001 дюйма) в английской системе и 0,01 мм в метрической.

В **P-CAD 6.0—8.7** з

- а счет перехода на 32-разрядную арифметику точность повышена на два порядка, что излишне,
- при этом система единиц задается в начале работы с проектом и в дальнейшем не может быть изменена.

В **P-CAD 2000 и 2001** в редакторе ПП установлена дискретность измерения линейных размеров 0,1 мил в английской и 0,001 мм в метрической системе, при этом за счет резерва точности внутреннего представления 32-разрядных данных обеспечивается возможность изменения системы единиц на любой стадии работы со схемой или ПП.

### **Преимущества ПП P-CAD 2001**

По сравнению с OLD PCAD для DOS система P-CAD для Windows имеет и другие преимущества:

- применение интерфейса в стиле Windows упрощает работу с системой и ознакомление с ней;
- обеспечивает вывод результатов проектирования на современные мониторы и периферийные устройства;
- решена проблема нанесения на схемы, ПП и фотошаблоны надписей по-русски;
- возможна тонкая настройка стратегии автоматической трассировки (задание разных типов переходных отверстий при переходе проводников со слоя на слой, трассировка одной и той же цепи сегментами разной ширины и др.);
- усовершенствованы алгоритмы автотрассировки проводников, включая трассировку многослойных ПП, имеющих внутренние слои металлизации;
- обеспечена координация библиотек символов и корпусов компонентов.

Поддержка текстовых форматов баз данных DXF, PDIF, ALT и др. позволяет обмениваться информацией с такими распространенными пакетами, как AutoCAD, OrCAD, P-CAD old, «Компас» и др.

Очередная версия P-CAD 2001 вышла в начале января 2001 г. Австралийская фирма Protel International внедрила в схемный редактор P-CAD Schematic интерфейс с собственной программой моделирования аналого-цифровых устройств Circuit Simulator, используемой в системе Protel 99 SE.

В комплект поставки P-CAD 2001 снова включена программа анализа паразитных эффектов ПП Signal Integrity фирмы INCASES Engineering (она входила в состав ранних версий ACCEL EDA 13—15, но была исключена из P-CAD 2000).

Еще одна программа, включенная в интерфейс редактора печатных плат P-CAD PCB, — программа доработки управляющих файлов для фотоплоттеров. Она корректно подготавливает управляющие файлы в формате Gerber и позволяет их отредактировать и доработать перед созданием фотошаблонов на технологическом оборудовании, заменяя аналогичную программу CAM350 фирмы Advanced CAM Technologies.

В P-CAD 2001 в состав программ автотрассировки включен бессеточный автотрассировщик Shape-Based Router, в котором применен алгоритм оптими-

зации нейронных сетей, заимствованный из системы Protel (аналогичная программа под названием SmartRoute используется и в системе OrCAD 9).

Однако интерфейс с хорошо зарекомендовавшей себя программой SPECSTRA сохранен.

Из менее значительных нововведений P-CAD 2001 отметим следующие:

- перед выводом на печать чертежей схем и ПП предусмотрена возможность задания минимальной ширины линий (отсутствие этого в предыдущих версиях P-CAD снижало качество схем, выводимых на лазерные печатающие устройства, — некоторые линии были слишком тонкими);
- в файл технологических параметров включена секция для хранения апертур фотоплоттера;
- в графический редактор ПП включена команда вставки точек изломов проводников, имевшаяся в OLD PCAD ;
- в графический редактор ПП включена команда маскирования, задающая режим выделения наложенных друг на друга объектов;
- в менеджере компонентов введена возможность одновременного просмотра состава нескольких библиотек.

### **Требования к оборудованию**

Система P-CAD 2001 устанавливается на ПК:

- процессор 486 и Pentium,
- работает под управлением Microsoft Windows 95/98/2000/NT
- поставляется на компакт-диске вместе с электронной документацией в формате Adobe Acrobat Reader.
- На жестком диске P-CAD 2001 вместе с библиотеками и утилитами занимает примерно 350 Мб (наибольший объем занимают библиотеки).
- Минимальный объем ОЗУ составляет 16 Мб, рекомендуется 32 Мб (при автотрассировке сложных ПП объем ОЗУ следует еще увеличить),
- минимальное разрешение монитора 800x600.

Для работы с P-CAD 2001 не требуется глубоких знаний в области вычислительной техники, достаточно иметь общее представление о ПК, операционной системе Microsoft Windows и желательно знать основы английского языка. **Главное — необходимо быть специалистом в области разработки ПП.**

## **8.2 Состав и структура системы P-CAD**

Система P-CAD 2001 предназначена для проектирования многослойных печатных плат (ПП) электронных устройств в среде Windows. Она состоит из четырех основных модулей:

1. P-CAD Library Manager (или Library Executive),

2. P-CAD Schematic, P-CAD PCB,
3. P-CAD Autorouters (рисунок 8.1) и
4. ряда вспомогательных программ.

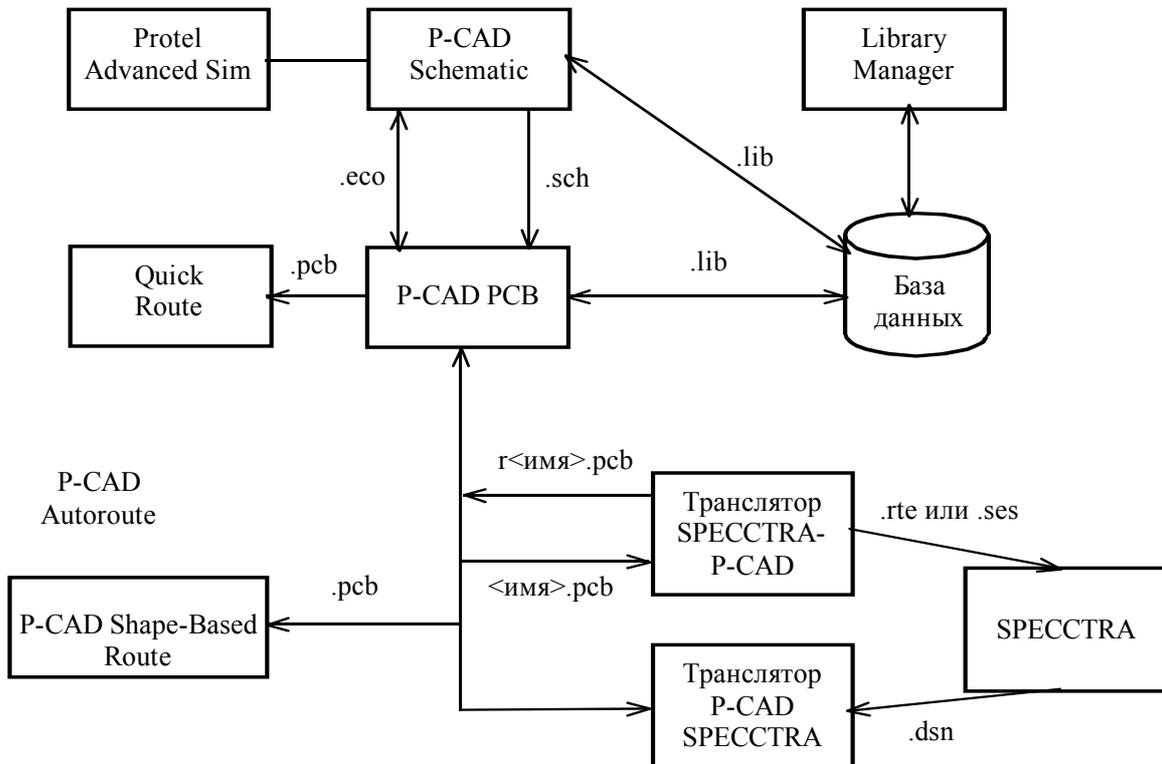


Рисунок 8.1 – Структура системы P-CAD

**1) P-CAD Library Manager (Library Executive)** — менеджер библиотек. Система P-CAD имеет интегрированные библиотеки, которые содержат графическую и текстовую информацию о компонентах. В графическом виде представлена информация о графике символов и корпусов компонентов; в текстовом виде - число секций в корпусе компонента, номера и имена выводов, коды логической эквивалентности выводов и секций и т. п.

(В old PCAD имелись отдельные библиотеки символов и корпусов, содержащих повторяющуюся текстовую информацию.)

Исключаются ошибки несогласованного ввода этих данных, возможные в old PCAD. Библиотеки символов и корпусов компонентов OLD PCAD 6.0—8.7 через текстовый формат PDIF переносятся по отдельности в P-CAD 2001 и затем объединяются в интегрированные библиотеки.

Утилита **Library Executive** состоит из программы **Library Manager**, в которую включен ряд дополнительных команд, и редакторов символов компонентов **Symbol Editor** и их корпусов **Pattern Editor**.

**2) P-CAD Schematic и P-CAD PCB** — графические редакторы схем и ПП.

В отличие от OLD PCAD для выполнения простейших операций перемещения, копирования, поворота или удаления объектов в P-CAD Schematic и P-CAD PCB не нужно пробираться через последовательность разных меню, это

делается гораздо проще, в стиле Windows. Например, для перемещения объекта сначала курсором включают режим выбора, отмечают щелчком *левой клавиши* мыши нужный объект и затем перемещают его движением мыши; поворот объекта при этом выполняется нажатием клавиши R, зеркальное отображение — клавиши F. *Двойной щелчок* левой клавишей мыши по выбранному объекту обеспечивает доступ к просмотру и редактированию объектов.

Копирование объектов в буфер обмена или в файл позволяет не только переносить их из одной базы данных в другую, но и помещать в другие программы Windows, например в MS Word для выпуска документации.

В поставляемых вместе с системой библиотеках импортных цифровых ИС некоторые компоненты имеют три варианта графики: Normal — нормальный (в стандарте США), DeMorgan — обозначения логических функций, IEEE — обозначения в стандарте Института инженеров по электротехнике и электронике (наиболее близком к отечественным стандартам).

В P-CAD 2001 PCB предусмотрена возможность создания альтернативной графики и для корпусов компонентов. Библиотеки отечественных компонентов выполняются по ЕСКД в среде P-CAD или переносятся из OLD PCAD. Средствами Windows реализован многооконный интерфейс, что разрешает на одном экране просмотреть чертежи схем и ПП и провести идентификацию на ПП цепей, выделенных на схеме (реализуя так называемую «горячую» связь). Применение шрифтов TrueType позволяет наносить на схемы и ПП надписи по-русски.

Графический редактор ПП P-CAD PCB вызывается автономно или из редактора схем P-CAD Schematic. В P-CAD Schematic составляется список соединений схемы (Netlist), который загружается в P-CAD PCB, и на поле ПП переносятся из библиотек изображения корпусов компонентов с указанием линий электрических соединений между их выводами — по терминологии OLD PCAD эта операция называется *упаковкой схемы на ПП*. После этого вычерчивается контур ПП, внутри него (вручную или в интерактивном режиме с помощью SPECCTRA) размещаются компоненты и проводится трассировка проводников.

В P-CAD PCB появилось **много новых возможностей**, позволяющих улучшить качество разработки ПП.

- средства обнаружения и удаления изолированных островков металлизации,
- автоматическая очистка зазоров в областях металлизации при прокладке через занятые ими области проводников и простановке переходных отверстий (ПО),
- возможность задания индивидуальных зазоров для разных проводников, классов проводников и проводников, находящихся на различных слоях или в различных областях (комнатах),
- расщепления металлизированных слоев на области для подключения нескольких источников питания, отдельно аналоговой и цифровой «земли».
- ПО допускается размещать в любой точке ПП, что облегчает разметку центров крепежных отверстий (в OLD PCAD для этого создаются фиктивные

компоненты, состоящие из одного вывода).

Еще одно отличие от OLD PCAD — изображения стеков контактных площадок (КП) и ПО не хранятся в виде отдельных графических файлов, подключаемых на заключительной стадии проектирования ПП, теперь они создаются в текстовом виде и хранятся вместе с базой данных ПП или в файле технологических параметров проекта (при этом имеется возможность их оперативной замены).

Базы данных схем и ПП всех версии OLD PCAD через текстовый формат PDIF переносятся в P-CAD Schematic и P-CAD PCB.

**3) Autorouters.** В состав P-CAD 2001 входят два автотрассировщика:

1. простейшая программа QuickRoute и
2. заимствованная из системы Protel программа Shape-Based Router (аналогичная программе SmartRoute из системы OrCAD 9.2).
3. программой SPECCTRA (поставляется отдельно)

Они вызываются из управляющей оболочки P-CAD PCB, в которой производится настройка стратегии трассировки. Очень удобно, что информацию об особенностях трассировки отдельных цепей можно с помощью стандартных атрибутов ввести еще на этапах создания принципиальной схемы или ПП:

- атрибуты ширины трассы,
- типа ассоциируемых с ней ПО и их максимально допустимого количества,
- признак запрета разрыва цепи в процессе автотрассировки,
- признак предварительно разведенной и зафиксированной цепи.

Эффективность и «интеллектуальность» алгоритмов трассировки Shape-Based Router гораздо выше, чем у OLD PCAD 8.7, но Shape-Based Router уступает программе SPECCTRA по возможностям тонкой настройки стратегии трассировки; к тому же SPECCTRA обладает возможностями размещения компонентов в интерактивном и автоматическом режимах.

**4.1) P-CAD Document Toolbox** — набор дополнительных команд для автоматизации создания графической и текстовой информации, необходимой для документирования результатов проектирования (вызов этих команд встраивается в набор инструментов программ P-CAD Schematic и P-CAD PCB):

для размещения на чертежах схем и ПП различных диаграмм, таблиц сверловки, составления списков соединений, выводов подключения питания и других текстовых отчетов, которые динамически обновляются.

**4.2) P-CAD InterPlace&PCS (Pametric Constraint Solver)** — DBX-утилита, данные в которую передаются из P-CAD Schematic или P-CAD PCB.

Модуль *Pametric Constraint Solver* позволяет задать набор правил размещения компонентов, трассировки проводников и других правил разработки ПП на этапах создания принципиальной схемы и ранних этапах работы с печатными платами. Эти данные передаются в программы Shape-Based Router и SPECCTRA. При задании правил разработки ПП допускается использовать математические функции.

Модуль *InterPlace* представляет собой интерактивное средство размещения

компонентов (базовая программа P-CAD PCB позволяет выполнить размещение компонентов только вручную, для автоматического размещения используется отдельная дорогостоящая программа SPECCTRA). Компоненты могут быть объединены в физические или логические группы и размещены на ПП в определенных областях (комнатах), выровнены, перемещены или повернуты.

**4.3) P-CAD InterRoute Gold** — лицензия, обеспечивающая доступ в P-CAD PCB к дополнительным командам интерактивной трассировки проводников **Route>Bus, Fanout, Multi Trace, Push Traces, Visible Routing Area** и опциям **Maximum Hugging, Minimum Length** (которые, однако, имеются и в программе SPECCTRA).

**4.4) P-CAD Relay** — средство для обеспечения коллективной работы над проектами ПП. Является аналогом графического редактора P-CAD PCB с ограниченными возможностями. Печатные платы можно просматривать, вручную редактировать и выполнять вывод на принтеры и плоттеры.

С ее помощью разработчик схем может выполнить общую расстановку компонентов на ПП, задать наиболее существенные атрибуты, которые будут использованы при трассировке (например, допустимые зазоры), и проложить наиболее критичные трассы. Затем эти результаты передаются конструктору для завершения разработки ПП с помощью P-CAD PCB. Кроме того, с помощью P-CAD Relay выполняется контроль соблюдения технологических норм DRC и запускаются утилиты DBX.

**Schematic Viewer, PCB Viewer** — средство просмотра схем и ПП.

**Protel Advanced Sim** — программа моделирования аналоговых и смешанных аналого-цифровых устройств фирмы Protel, интегрируемая с P-CAD Schematic.

**P-CAD Signal Integrity** — программа анализа электрических характеристик ПП с учетом паразитных параметров реальных конструкций (задержки сигналов в печатных проводниках, учет потерь в них, наличия индуктивностей и емкостей, взаимных индуктивностей, согласование печатных проводников с источниками сигналов и нагрузками).

**SPECCTRA** — программа ручной, интерактивной и автоматической трассировки проводников и размещения компонентов фирмы Cadence Design Systems ([www.cadence.com](http://www.cadence.com)). Программа SPECCTRA успешно трассирует ПП большой сложности (число слоев до 256) благодаря применению так называемой бессеточной (Shape-Based) технологии. В отличие от разработанных ранее сеточных трассировщиков, в которых графические объекты представлены в виде набора координат точек, в ней используются компактные способы их математического описания. За счет этого повышается эффективность трассировки ПП с высокой плотностью расположения компонентов и обеспечивается тонкая настройка сложных стратегий размещения компонентов и трассировки проводников.

Всем объектам ПП присваивается определенный уровень иерархии и вводятся правила размещения и трассировки, составляющие их *стратегию*, предусматривающую особенности разработки конкретной ПП.

Помимо обычного контроля соблюдения технологических зазоров типа проводник — проводник, проводник — ПО и т. п. в системе SPECCTRA можно выполнить контроль максимальной длины параллельных проводников, расположенных на одном или двух смежных слоях, что позволяет уменьшить уровень перекрестных искажений. Контролируется также максимальное запаздывание сигнала в отдельных цепях.

SPECCTRA воспринимает информацию о ПП, подготовленных с помощью одного из графических редакторов: OLD PCAD , P-CAD PCB, MicroSim PCBboards, PADS, Protel, OrCAD Layout и многих др. Разработанная в SPECCTRA ПП транслируется обратно для выпуска документации. Процедура такой двусторонней трансляции встроена в P-CAD PCB, но она может выполняться и автономно.

**Вспомогательные утилиты.** Система P-CAD поставляется совместно с большим количеством утилит, написанных на Visual Basic, Си или Си++ и образующих интерфейс DBX (Data Base Exchange). Эти утилиты извлекают данные из открытых файлов схем или ПП, обрабатывают их, передают третьим программам и вносят коррективы в текущую базу данных. В частности, с их помощью можно производить изменение нумерации компонентов, создавать отчеты в заданном пользователем формате, автоматически создавать символы и конструктивы компонентов, рассчитывать паразитные параметры ПП и т. п.

**Программа доработки фотошаблонов SAMtastic.** Графический редактор P-CAD PCB создает управляющие файлы для фотоплоттеров типа Gerber и сверлильных станков с ЧПУ, причем управляющие файлы для фотоплоттеров не всегда корректны. Поэтому перед изготовлением фотошаблона целесообразно просмотреть и при необходимости отредактировать его изображение с помощью программы SAMtastic, интерфейс с которой встроен в P-CAD PCB (передача данных всегда производится корректно).

Для изготовления фотошаблонов на отечественном оборудовании необходимо перекодировать управляющие файлы.

### **Основные характеристики системы P-CAD. Общие характеристики:**

- 32-разрядная база данных;
- не более 20 000 компонентов в одной библиотеке;
- неограниченное количество компонентов в проекте;
- не более 64 000 цепей в проекте;
- не более 999 выводов в компоненте;
- не более 255 секций в компоненте;
- не более 2000 символов в атрибуте или в текстовой строке;
- не более 20 символов в имени вывода, имени цепи, позиционном обозначении вывода (пробелы, знаки табуляции, точки и скобки не допускаются);
- не более 16 символов в имени типа компонента;
- не более 30 символов в позиционном обозначении компонента (символы кириллицы допускаются, но не рекомендуются; символы двоеточия, пробелы,

знаки табуляции, точка и точка запятой не допускаются);

- минимальный шаг сетки 0,1 мил в английской системе и 0,001 мм = 1 мкм в метрической системе (1 мил = 0,001 дюйма = 0,0254 мм). Систему единиц можно изменять на любой фазе работы с проектом;

- автоматическая коррекция принципиальных схем по изменениям в ПП и наоборот (коррекция «назад» и «вперед»).

### **Графический редактор схем P-CAD Schematic:**

- не более 99 листов схем в проекте, максимальный размер листа 60x60 дюймов;

- поддержка стандартных форматов A—E, AO—A4 и форматов, задаваемых пользователями;

- дискретность угла поворота графической информации и текста 90° (поворачивать текст на 180° не допускается, для этого - специальные шрифты);

- составление списков соединений схем в форматах P-CAD ASCII, OLD PCAD, Tango, FutureNetEDIF 2.0.0 и PSpice, XSpice;

- контроль ошибок в принципиальных схемах;

- перекрестные связи между P-CAD Schematic и P-CAD PCB позволяют для выбранной на схеме цепи высветить на ПП соответствующий ей проводник и наоборот (так называемая горячая связь);

### **Графический редактор печатных плат P-CAD PCB:**

- не более 99 слоев в базе данных ПП;

- максимальный размер ПП 60x60 дюймов;

- не более 64 000 типов контактных площадок в проекте;

- ширина трассы не более 1 см (394 мил);

- min дискретность угла поворота графических объектов и текста 0,1°;

- неограниченное количество вершин в полигоне;

- не более 64 000 стилей стеков контактных площадок в проекте;

- контактные площадки 12 форм: эллипс, овал, прямоугольник, скругленный прямоугольник, полигон, непосредственное соединение со слоем металлизации, тепловой барьер с 2 или 4 перемычками 2-х ориентации каждое, перекрестье для сверления (Target), крепежное отверстие (Mounting Hole);

- контроль соблюдения зазоров и полноты разводки ПП;

- поддержка управляющих файлов фотоплоттеров Gerber (форматы RS-274-D и RS-274-X), сверлильных станков с ЧПУ типа Excellon (формат настраивается пользователем);

- передача данных программе анализа целостности сигналов Signal Integrity.

### **Размерности величин**

При работе в системе P-CAD приходится задавать значения разнообразных величин: ширины линий, зазоров, шагов сетки и т. п. Численное значение этих

величин может сопровождаться суффиксом для обозначения единицы измерения: mm — миллиметр, mil — мил (0,001 дюйма), inch — дюйм. Суффикс указывается вслед за числом без пробела, например 40mil, 0.25mm. Если суффикс опущен, величина понимается в текущей системе единиц. Однако в связи с тем, что в P-CAD систему единиц можно изменить на любой стадии работы с проектом, для предотвращения ошибок рекомендуется всегда указывать суффиксы размерности.

### 8.3 Создание библиотеки компонентов в САПР PCAD

САПР PCAD представляет собой пакет специализированных модулей, тесно связанных друг с другом и охватывающих все этапы разработки и изготовления однослойных и многослойных печатных плат.

Программные средства системы позволяют автоматизировать весь процесс проектирования электронных средств, начиная с ввода принципиальной схемы, ее моделирования, упаковки схемы на печатную плату, интерактивного размещения радиоэлектронных компонентов на плате и автотрассировки соединений, вплоть до получения конструкторской документации и подготовки информации для производства плат на технологическом оборудовании.

Поставляемые с системой PCAD библиотеки не соответствуют белорусским стандартам, поэтому для успешного проектирования печатных плат важно уметь создавать новые библиотечные компоненты и редактировать старые.

Интегрированные библиотеки PCAD 2001 содержат компоненты (components), корпуса (pattern) и символы (symbol). На схеме компонент представлен символом, а на печатной плате корпусом. Кроме графики символа и корпуса в библиотеке содержится информация об упаковке в корпус (подвод питания, подключение выводов и т.д.). Единство символа, графики корпуса и упаковочной информации и составляет понятие компонента. Преимущество интегрированных библиотек заключается в том, что упаковочная информация для каждого компонента хранится в одном месте и должна вводиться всего один раз.

Поэтому при создании библиотеки компонентов в PCAD 2001 необходимо:

- 1) создать новую библиотеку;
- 2) в редакторе Symbol Editor создать символ компонента и сохранить его в библиотеку под своим именем;
- 3) в редакторе Pattern Editor создать корпус компонента и сохранить его в библиотеку под своим именем;
- 4) в блоке Library Executive произвести объединение (упаковку) корпуса и символа в один компонент.

## Редактор символов Symbol Editor

Создавать символы компонентов можно тремя различными способами:

- 1) в редакторе PCAD Schematic;
- 2) в редакторе PCAD Symbol Editor;
- 3) путем редактирования существующего символа компонента.

Наиболее удобно создавать символы компонентов в редакторе PCAD Symbol Editor.

Редактор PCAD Symbol Editor имеет аналогичный с основным редактором PCAD Schematic экран (рисунок 8.2). Он имеет набор команд, необходимых для редактирования символов компонентов и аналогичных соответствующим командам PCAD Schematic за исключением команд меню **File**. В PCAD Symbol Editor работают с файлами библиотек (\*.LIB) и отдельных символов (\*.SYM). Система единиц, набор толщин линий и ряд других глобальных параметров

PCAD Symbol Editor сохраняются в файле конфигурации SymEd.INI: например, система единиц устанавливается с помощью ключевого слова Units, принимающего значения 0 — mil, 1 — inch, 2 — мм. По умолчанию устанавливается сетка 100 mil или 2,54 мм, поэтому необходимо предварительно настраивать параметры редактора в меню **Options**.

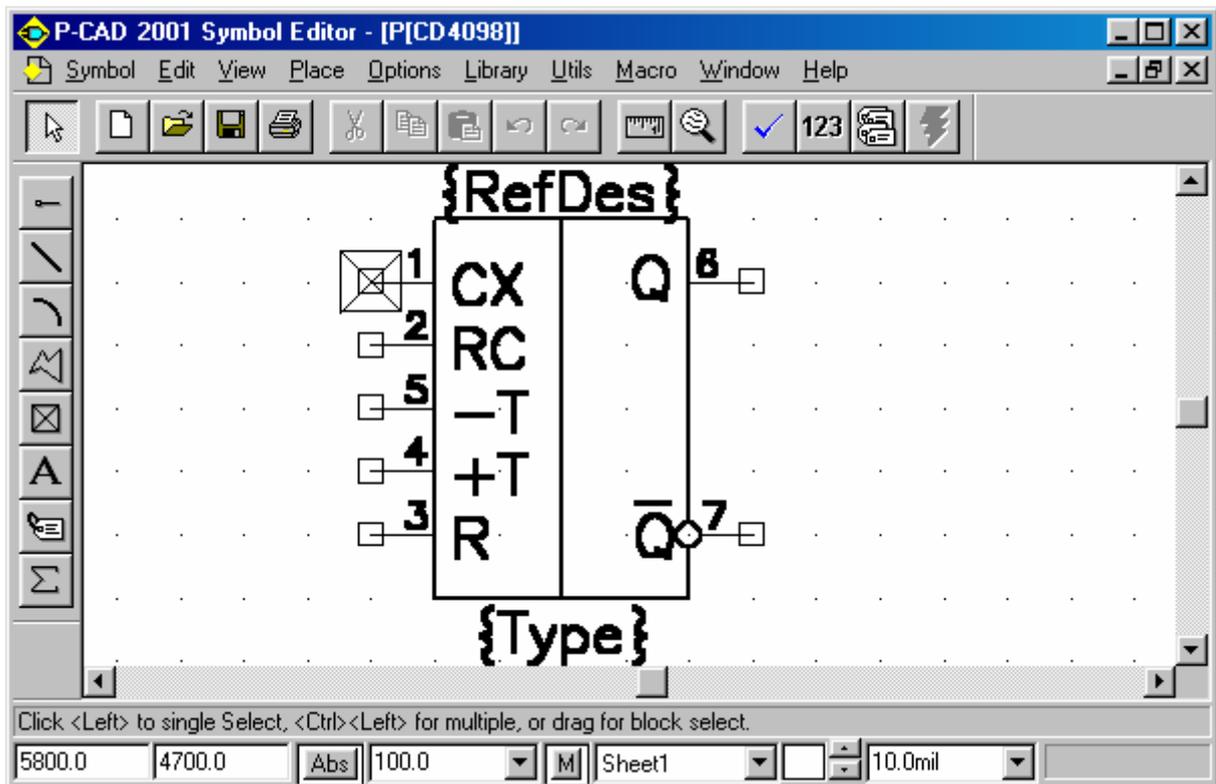


Рисунок 8.2 – Диалоговое окно Symbol Editor

При создании символов компонентов с большим количеством выводов удобен «Мастер Символов», вызываемый по команде **Symbol Wizard** меню

**Symbol** или кнопкой . В его диалоговом окне (рисунок 8.3) указывается следующая информация: **Symbol Width** — ширина символа; **Pin Spacing** — расстояние между смежными выводами; **Length** — длина вывода (Short, Normal, Long, User); **Number Pin Left (Right)** — количество выводов на левой (правой) стороне символа; **Symbol Outline** — необходимость изображать контур символа; **Line Width** — ширина линии контура символа; **Display Pin Name (Pin Des)** — необходимость указывать на чертеже символа имена (номера) выводов; **Default Pin Name** — имя первого вывода, принимаемое по умолчанию (не более 20 символов); **Default Pin Designator** — номер первого вывода, принимаемый по умолчанию; **Current Pin Number** — порядковый номер текущего вывода.

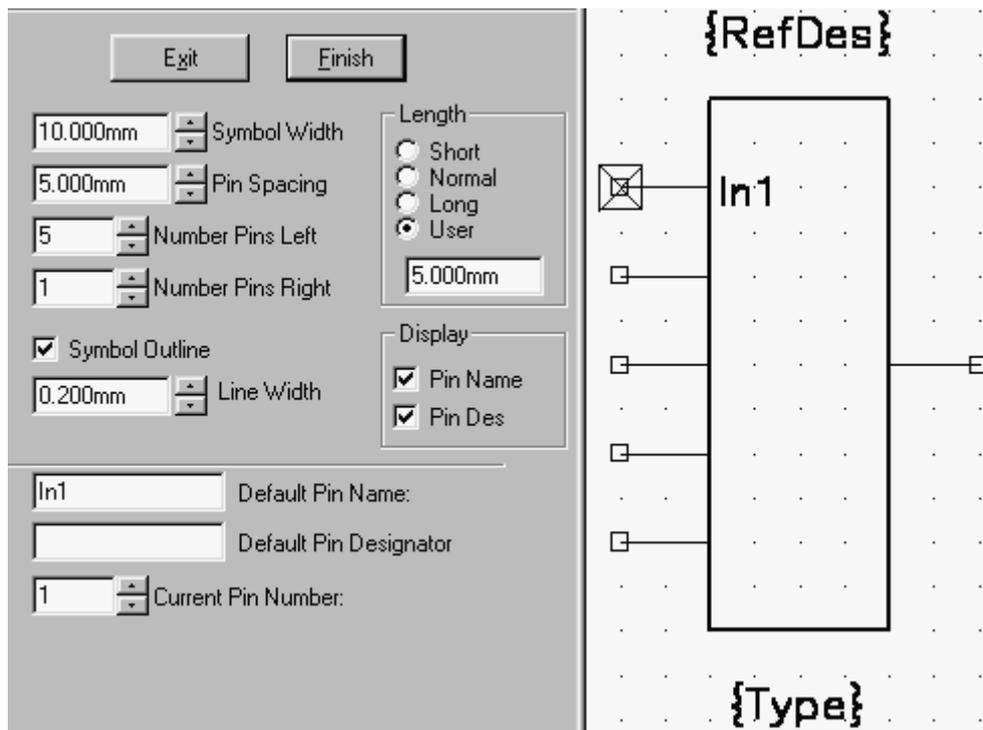


Рисунок 8.3 – Диалоговое окно Symbol Wizard

Нажатие на клавишу **Finish** завершает создание символа и его изображение заносится в библиотеку по команде **Symbol>Save** или **Symbol>Save As**. В связи с тем, что Мастер всегда создает прямоугольный символ без разбиения на зоны, графику символа обычно требуется редактировать.

При ручном рисовании символа полезна команда **Place>Pin**, диалоговое окно которой (рисунок 8.4) в удобной форме содержит всю информацию, необходимую для настройки режима размещения выводов символов, и окно для просмотра их графики.

В меню **Utils** (см. рис. 8.2) включена команда **Validate**, предназначенная для проверки корректности создания символа компонента: при наличии ошибок такой символ не разрешается заносить в библиотеку.

С помощью PCAD Symbol Editor вводятся атрибуты символов, сохраняемые в библиотеках. Эти атрибуты вводятся и редактируются по команде

**Symbol>Attributes**  или по команде **Place>Attribute** . Удобнее вводить атрибуты с помощью PCAD Symbol Editor, а не с помощью PCAD Schematic, так как при создании библиотеки компонентов с однотипными атрибутами проще их ввести один раз, копировать вместе с символом и затем редактировать.

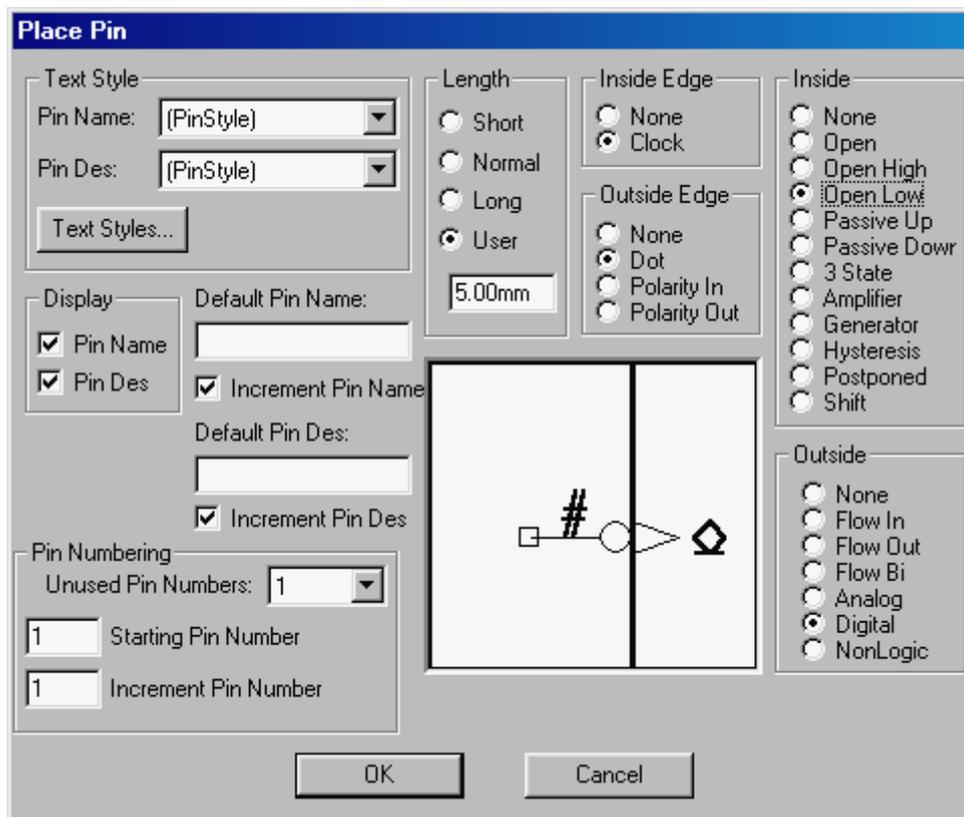


Рисунок 8.4 – Диалоговое окно Place Pin

Для копирования символа он сначала загружается по команде **Symbol>Open** (если символ находится не в отдельном файле, а в библиотеке, предварительно указывается ее имя). После внесения изменений в атрибуты символ сохраняется по команде **Symbol>Save As**. Если отметить при этом опцию **Create Component**, созданный символ можно сразу же наносить на схему без описания корпуса и информации об упаковке элемента.

### Редактор корпусов Pattern Editor

Создавать корпуса компонентов можно тремя различными способами:

- 1) в редакторе PCAD PCB;
- 2) в редакторе PCAD Pattern Editor;
- 3) путем редактирования существующего корпуса компонента.

Наиболее удобно создавать символы компонентов в редакторе PCAD Pattern Editor.

При создании корпуса компонента необходимо помнить, что создается проекция корпуса на плату с формованными выводами. В общем случае вид проекции корпуса может значительно отличаться от внешнего вида корпуса

компонента.

Экран редактора корпусов компонентов PCAD Pattern Editor (рисунок 8.5) в основном такой же, как экран основного редактора PCAD PCB. PCAD Pattern Editor имеет набор команд, необходимых для редактирования корпусов компонентов. В PCAD Pattern Editor работают с файлами библиотек (\*.LIB) и отдельных корпусов (\*.PAT).

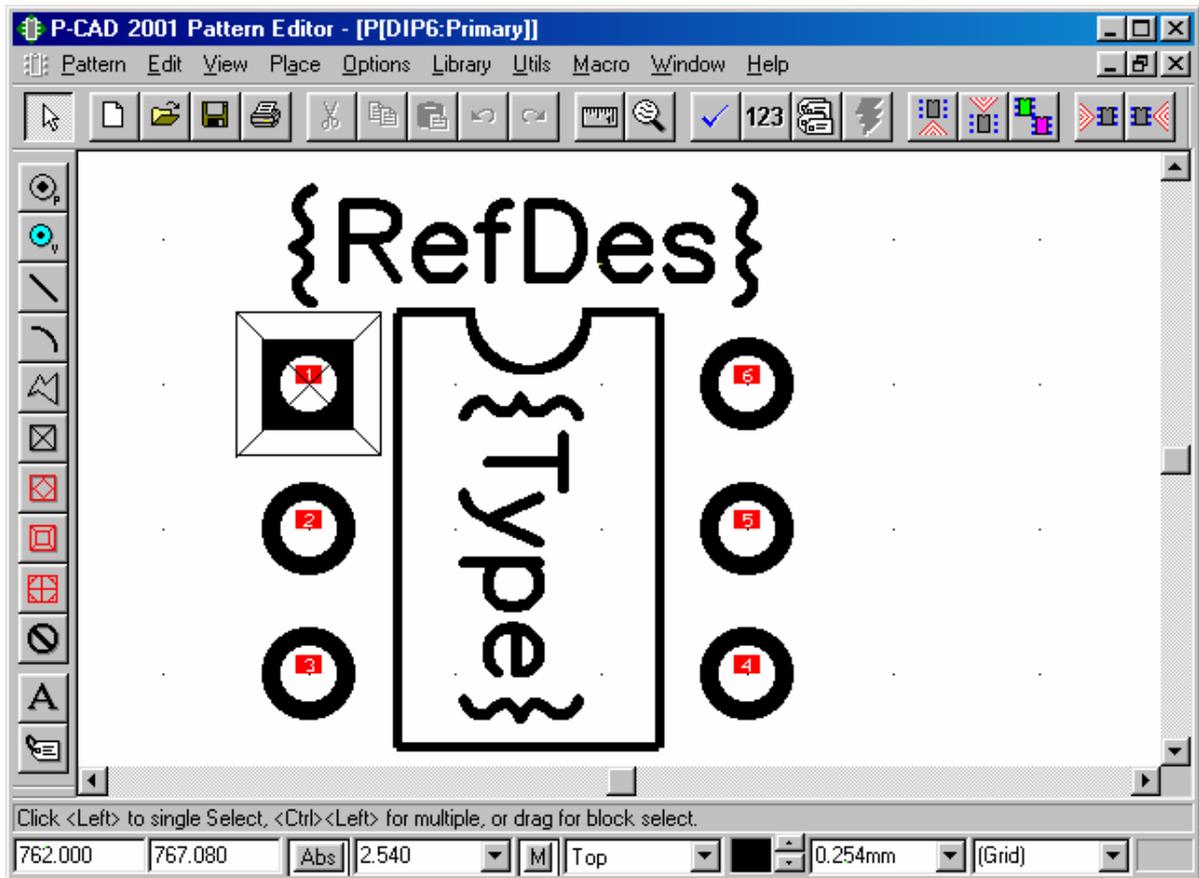


Рисунок 8.5 – Экран PCAD Pattern Editor

В PCAD Pattern Editor параметры конфигурации сохраняются в файлах отдельных символов «\*.PAT». Поэтому можно установить необходимый набор сеток и сохранить «пустой» файл шаблона корпуса по команде **File>Copy To File As**. Система единиц и ряд других глобальных параметров сохраняются в файле конфигурации PatEd.INI. Перед вызовом программы PCAD Pattern Editor устанавливается нужный файл конфигурации и по команде **File>Open** загружается шаблон корпуса.

По команде **Pattern Wizard** или кнопкой  вызывается «Мастер создания корпусов компонентов». В его диалоговом окне (рисунок 8.6) указывается следующая информация: **Pattern Type** - тип корпуса (посадочного места) компонента; **Number of Pads Down** - число строк выводов; **Number of Pads Across** - число столбцов в массиве выводов; **Pad to Pad Spacing (On Center)** - расстояние между центрами выводов; **Pattern Width** - расстояние между крайними столбцами выводов (для корпусов DIP, QUAD); **Pattern Heigh** - расстояние ме-

жду крайними строками выводов (для корпуса QUAD); **Pad 1 Position** - расположение первого вывода (для корпусов DIP, QUAD); **Pad Style** - тип стека контактной площадки (КП) (отдельно для первого и остальных выводов); **Silk Screen** - необходимость изображения контура корпуса; **Silk Line Width** - ширина линий контура корпуса; **Silk Rectangle Pattern Width** - ширина корпуса компонента; **Silk Rectangle Pattern Height** - высота корпуса компонента; **Rotate** - признак поворота контактных площадок на 90°; **Notch Type** - тип скоса графики корпуса компонента (в верхнем левом углу, в нижнем левом углу и т.п.).

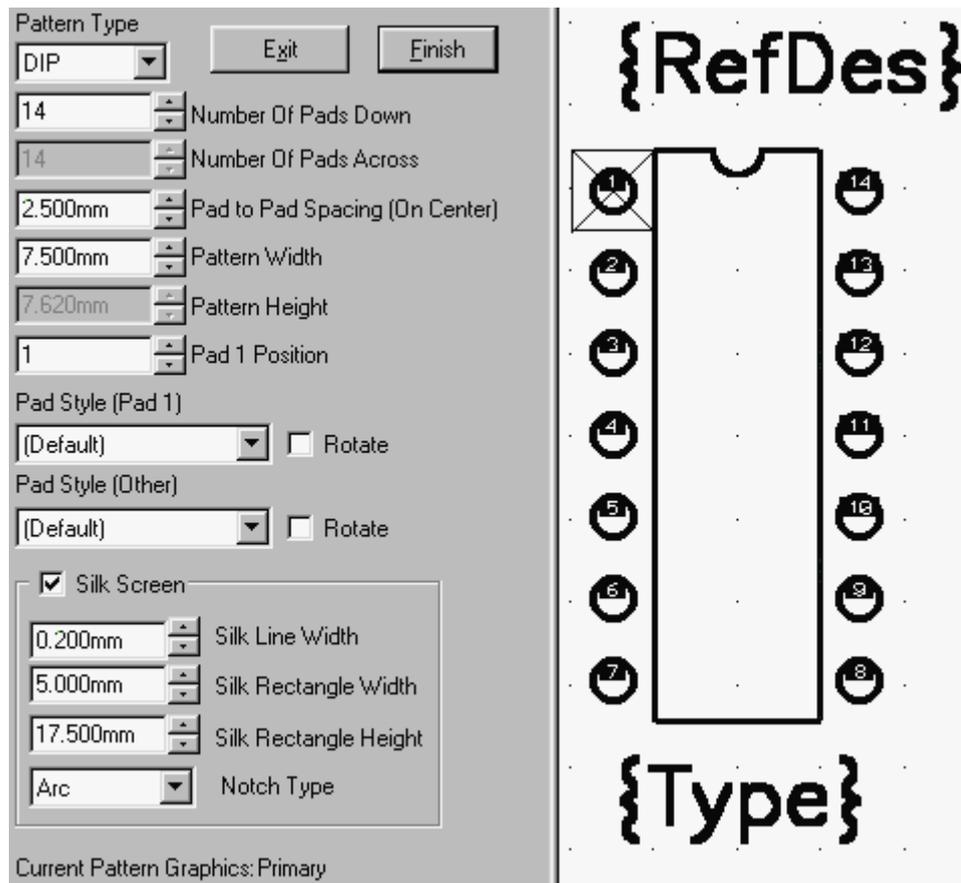


Рисунок 8.6 – Диалоговое окно PCAD Pattern Wizard

Нажатие на клавишу **Finish** завершает создание корпуса компонента, после чего его изображение переносится на основной экран программы Pattern Editor. Здесь его графику можно обычным способом отредактировать и затем занести в библиотеку по команде **Pattern>Save** или **Pattern>Save As**.

Одному символу компонента может соответствовать несколько корпусов (DIP, SMT, flat-pack и т.п.). По команде **Pattern>Add Pattern Graphics**  к основному изображению графики корпуса компонента (оно по умолчанию носит название Primary) добавляется альтернативное (рисунок 8.7). Назначение альтернативных корпусов выполняется в программе Library Executive по команде **Library>Pattern Graphics**.

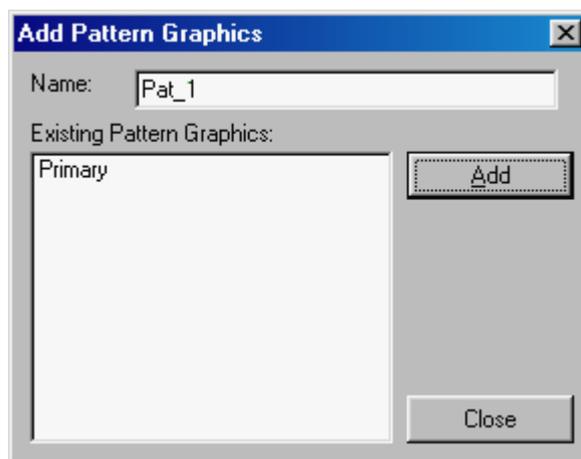


Рисунок 8.7 – Добавление альтернативного корпуса компонента

При редактировании существующего корпуса в рабочее окно редактора помещают изображение корпуса, хранящееся в библиотеке как единый объект. Затем его преобразуют в набор графических примитивов, что дает возможность внесения изменений и дополнений. После модификации корпус заносится в библиотеку под новым именем и сохраняется как единое целое. Последовательность действий такова.

1. Нажатием пиктограммы  включают режим выбора объектов и указывают редактируемый корпус щелчком левой кнопки мыши.

2. По команде **Edit>Explode Component** выбранный корпус преобразуется в набор графических примитивов — если до выполнения этой команды корпус представлял собой единое целое, то теперь доступны для редактирования все его составные части (выводы, линии, надписи и т.п.). Если добавлены или удалены выводы компонента, то нужно заново их пронумеровать по команде **Utils>Renumber**. Для этого перед выполнением команды включают режим выбора объектов, а в меню команды указывают режим перенумерации выводов **Pad Number**. После закрытия меню команды все выводы, подлежащие перенумерации, по очереди помечают щелчком мыши в порядке возрастания их номеров, начиная с первого (перенумерованные выводы окрашиваются).

3. После внесения всех изменений корпус заносят в библиотеку. Сначала выполняют операцию блочного выбора всех принадлежащих ему графических объектов, заключая их в прямоугольную рамку. Далее по команде **Library>Pattern Save As** заносят корпус в библиотеку. В списке **Library** выбирают имя одной из открытых библиотек и в графе **Pattern** вводят имя нового корпуса. Если компонент в целом будет создаваться в дальнейшем с помощью **Library Executive** или **Library Manager**, не нужно включать опцию **Create Component**.

### Менеджер библиотек компонентов

В системе PCAD поддерживаются два типа библиотек:

1) интегрированные библиотеки компонентов;

2) отдельные библиотеки символов и корпусов компонентов.

В интегрированную библиотеку заносятся данные трех типов:

- 1) текстовая информация о компонентах;
- 2) графика корпусов;
- 3) графика символов компонентов.

Способы создания графики корпусов и символов указаны выше. Менеджер библиотек Library Manager или Library Executive заносит эти данные в интегрированную библиотеку и добавляет текстовую информацию о компонентах.

Использование интегрированные библиотеки при проектировании печатных плат в PCAD предпочтительнее, так как это позволяет выполнять:

- “горячую связь” между графическими редакторами;
- прямую и обратную корректировку проекта;
- перестановку логически эквивалентных выводов и секций компонентов.

Содержание загруженных библиотек просматривают в графических редакторах по команде **Place>Part** или **Place>Component**, а также в менеджере библиотек по команде **Component>Open**.

Library Executive имеет ряд дополнительных средств по сравнению с Library Manager:

- включена команда Query для поиска компонентов в библиотеках по заданному набору атрибутов;
- при использовании Library Executive возможно применение редактора символов Symbol Editor и редактора корпусов Pattern Editor.

Упаковка компонентов в Library Executive – самый трудоемкий и ответственный этап в создании библиотечного компонента. Неправильное заполнение таблицы упаковки ведет к ошибочной трассировке печатной платы и даже к невозможности ее проектирования.

После загрузки программы Library Executive на строке инструментов доступны только пиктограммы **Component>New**, **Component>Open** и **View>Source Browser**  (недоступные пиктограммы и строки меню окрашены в серый цвет, при вызове **Source Browser** щелчком правой кнопки мыши открывается выпадающее меню, содержание которого зависит от типа объекта, выбранного в окне). После загрузки существующего компонента или открытия нового по командам **Component>Open**, **Component>New** на экране появляется диалоговое окно **Component Information**, показанное в левом верхнем углу рисунка 8.8.

На трех остальных окнах этого рисунка представлена детальная информация о компоненте, позволяющая установить связь между выводами символа и корпуса компонента. Всего меню Library Executive имеет четыре основных окна.

**1. Окно Component Information.** В этом окне представлена общая информация о компоненте:

- в строке Select Pattern выбирается тип корпуса компонента, например SO 14;

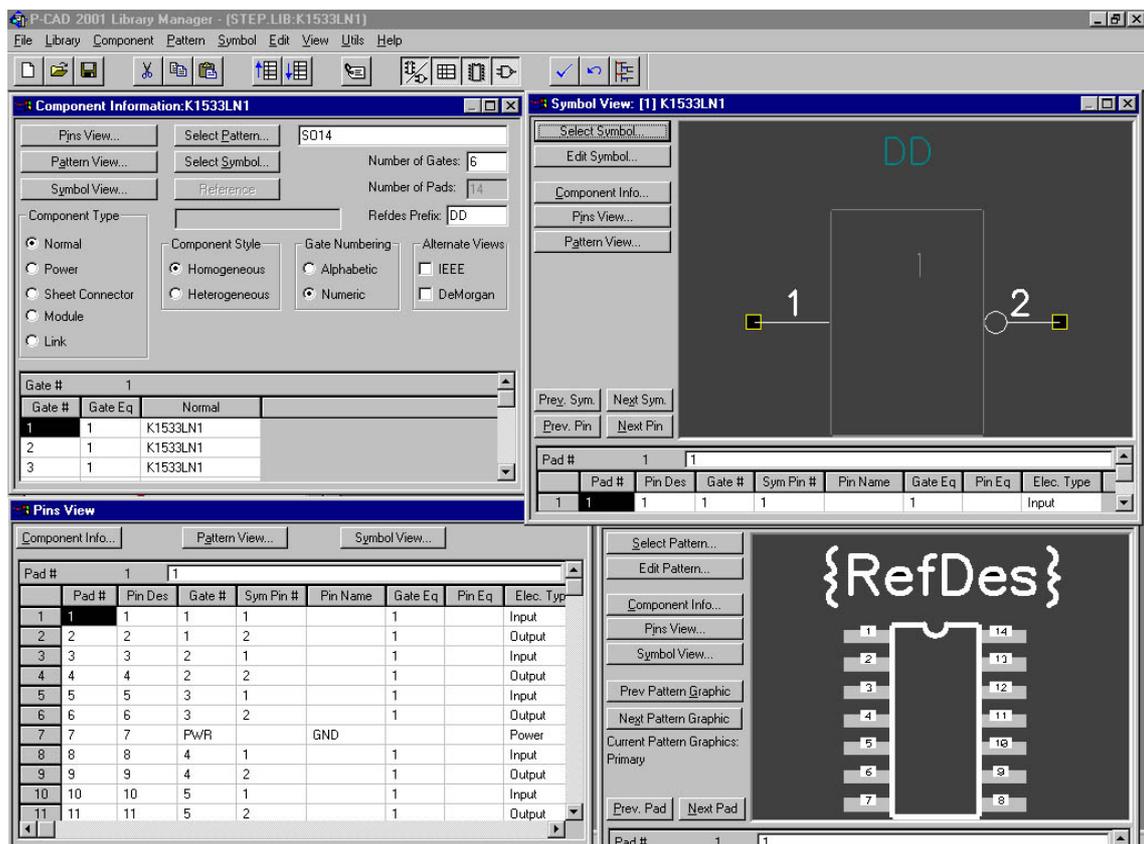


Рисунок 8.8 – Подробная информация о компоненте на экране Library Manager

- в строке Number of Gates — задается число секций в компоненте;
- в строке Number of Pads — указывается общее число выводов;
- в строке Refdes Prefix — задается префикс позиционного обозначения компонента;
- в разделе Component Type выбирают тип компонента (который принимается во внимание при составлении списков соединений и заполнении граф отчетов о проекте):

*Normal* — обычный компонент; *Power* — источник питания (компоненты такого типа, в частности, не включаются в списки соединений для «упаковки» схемы на ПП, но включаются в списки соединений для выполнения моделирования); *Sheet Connector* — соединитель листов схемы; *Module* — символ (модуль) иерархической структуры; *Link* — связь символа модуля иерархической структуры с его схемой;

- в разделе Component Style выбирают:

*Homogeneous* — однородный компонент (все секции однотипны);

*Heterogeneous* — неоднородный компонент (секции разных типов);

- в разделе Gate Numbering задают способ нумерации секций:

*Alphabetic* — буквенный; *Numeric* — числовой (рекомендуется);

- в разделе Alternate Views указывают альтернативные изображения символов.

В нижней части окна располагается таблица, где для каждого типа секций указаны:

Gate # — номер секции;

Gate Eq — код логической эквивалентности секции (секции, имеющие одинаковый отличный от нуля код эквивалентности, могут переставляться при оптимизации цепей на печатной плате);

Normal — имя символа в нормальном изображении.

Альтернативные обозначения символов можно использовать для введения их изображений по ЕСКД.

Вверху располагаются кнопки **Pins View**, **Pattern View** и **Symbol View** для открытия окон редактирования параметров выводов компонентов.

**2. Окно Symbol View.** В нем приведено изображение символа текущего компонента и таблица с информацией о его выводах. Каждому выводу компонента отведена одна строка. В столбцах указана следующая информация:

Pad # — номер вывода компонента;

Pin Des — физический номер вывода компонента в корпусе;

Gate # — номер секции компонента;

Sym Pin # — порядковый номер вывода символа компонента в пределах секции;

Pin Name — имя вывода символа секции компонента;

Gate Eq — код логической эквивалентности секции компонента;

Pin Eq — код логической эквивалентности вывода секции;

Elec. Type — электрический тип вывода, необходимый для проверки принципиальной схемы:

*Unknown* - неизвестный; *Passive* - вывод пассивного компонента; *Input* - вход; *Output* - выход; *Bidirectional* - двунаправленный вывод; *Open-H* - вывод секции с открытым эмиттером; *Open-L* - вывод секции с открытым коллектором; *Passive-H* - вывод пассивного компонента, подключенный к источнику высокого потенциала; *Passive-L* - вывод пассивного компонента, подключенный к источнику низкого потенциала; *3-State* - 3-стабильный вывод; *Power* - вывод цепи питания.

Нажатие на кнопку **Select Symbol** позволяет изменить символ, назначенный текущему компоненту.

**3. Окно Pattern View.** В нем приведено изображение корпуса текущего компонента и таблица с информацией о его выводах. Нажатие на кнопку **Select Pattern** позволяет изменить корпус, назначенный текущему компоненту.

**4. Окно Pins View.** В нем приведена таблица с информацией о всех выводах компонента, наиболее удобная для редактирования. После выбора курсором какой-либо ячейки выше таблицы появляется строка с информацией о занесенных в нее данных и панель для их редактирования.

Каждый компонент библиотеки состоит из одной или нескольких логических *секций* (gates), которые упаковываются в корпус. Несколько разных компонентов могут упаковываться в один и тот же типовой корпус

Редактор PCAD PCB не может использовать информацию только о компоненте (так как это только текстовая информация) или только о корпусе (так

как это только графическая информация), для него необходимы совместные данные компонент/корпус. Аналогично для редактора PCAD Schematic необходимы совместные данные компонент/символ. Когда компонент размещается на ПП или на схеме, используется графика корпусов или символов, на которые сделаны ссылки в описании компонента. Разные компоненты могут ссылаться на одну и ту же графику корпусов или символов. Корпуса и символы, на которые имеются ссылки в компонентах, должны находиться в одной и той же библиотеке.

Ссылки в разных компонентах на одни и те же корпуса и символы значительно экономят память, занимаемую библиотекой. Кроме того, редактирование типовых корпусов и символов сразу вносит изменения в графику всех родственных компонентов.

При работе с библиотекой компонентов важно различать следующие основные понятия (рисунок 8.9):

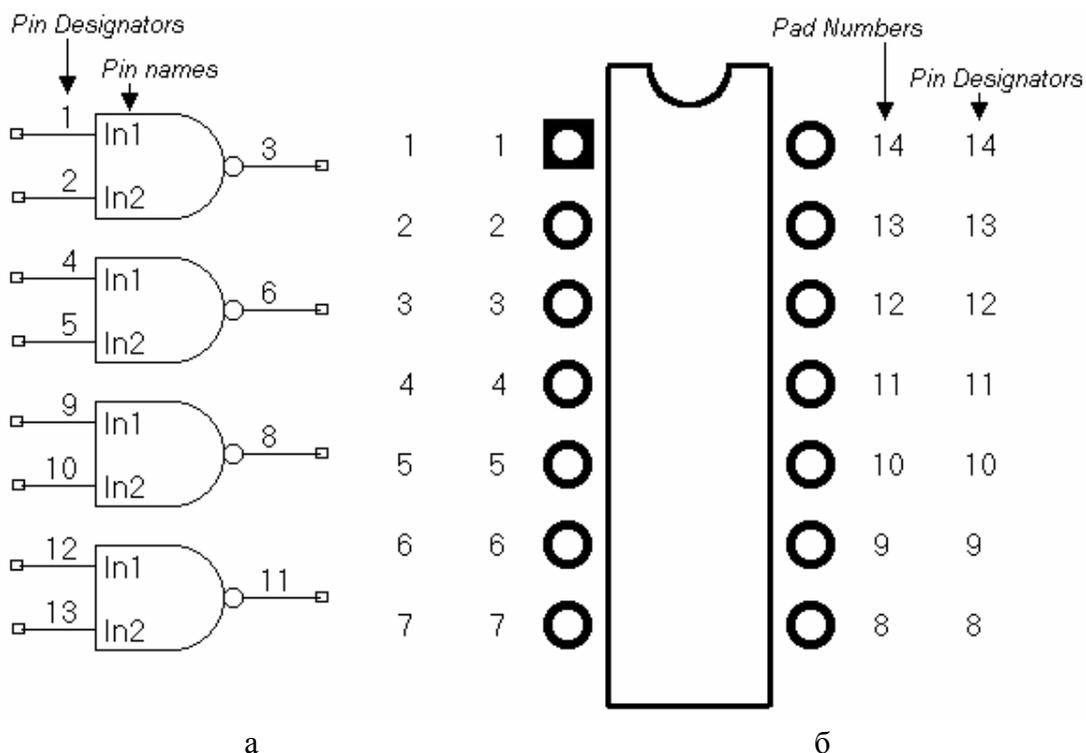


Рисунок 8.9 – Изображение компонента на схеме (а) и на ПП (б) ИС К155ЛА3

**Pad Number** — порядковый номер вывода компонента (обычно совпадает с физическим номером вывода, но это необязательно);

**Pin Designator** — физический номер вывода в корпусе;

**(Symbol) Pin Number** — номер вывода в пределах секции компонента;

**Pin Name** — имя вывода (в секции компонента).

Работа с библиотеками значительно облегчается с помощью окна просмотра **Source Browser** и команды поиска **Query**.

Окно просмотра **Source Browser** (рисунок 8.10) открывается по команде

**View>Source Browser.** В нем отображается дерево библиотек, открытых по команде **Library>Setup**. Каждая библиотека состоит из разделов Components, Patterns и Symbols. Двойной щелчок курсора мыши по символу или компоненту открывает окно просмотра их изображения. Щелчок правой кнопки мыши по компоненту открывает всплывающее меню, содержащее две строки: **Open** и **Place**. Выбор варианта Open открывает окно Component Information с предоставлением возможностей редактирования параметров компонента. Выбор варианта **Place** позволяет по дополнительному выбору (PCB, Schematic) разместить символ компонента на схеме или его корпус на ПП (для этого предварительно должна быть открыта программа PCAD PCB или Schematic).

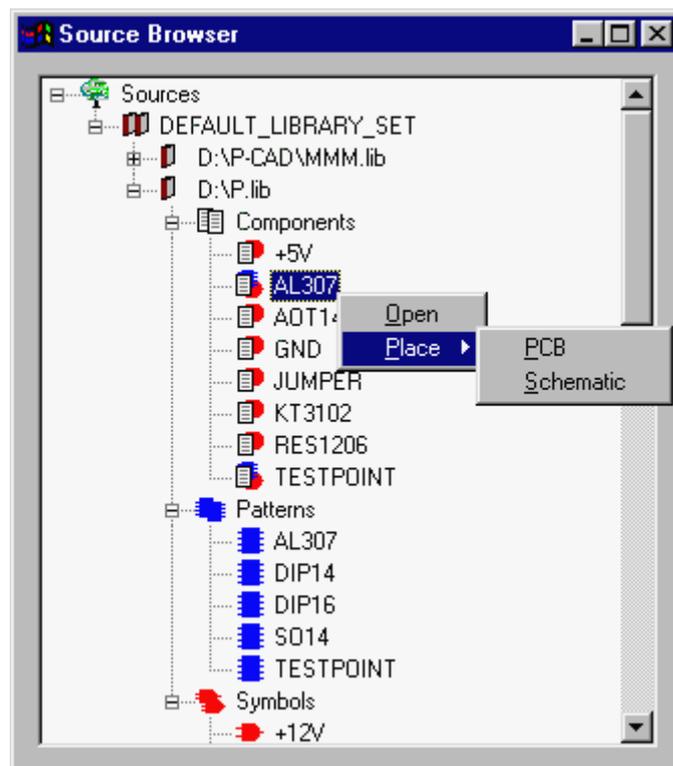


Рисунок 8.10 – Окно просмотра Source Browser

Изображение иконки компонента в окне просмотра свидетельствует о наличии у него присоединенного корпуса и/или символа.

### Создание компонента с помощью Library Executive

После загрузки Library Executive выполняется команда **Component>New**, и в ее диалоговом окне указывают файл библиотеки, в который ранее занесены корпус и символ компонента. Затем на экран выводится окно **Component Information** (рисунок 8.11). В нем сначала нажимают кнопку **Select Pattern** для подключения графики корпуса компонента. В открывшемся окне **Library Browse** из списка корпусов, помещенных в открытую библиотеку, выбирают нужный - в рассматриваемом примере это SO 14. Затем вводят следующую информацию.

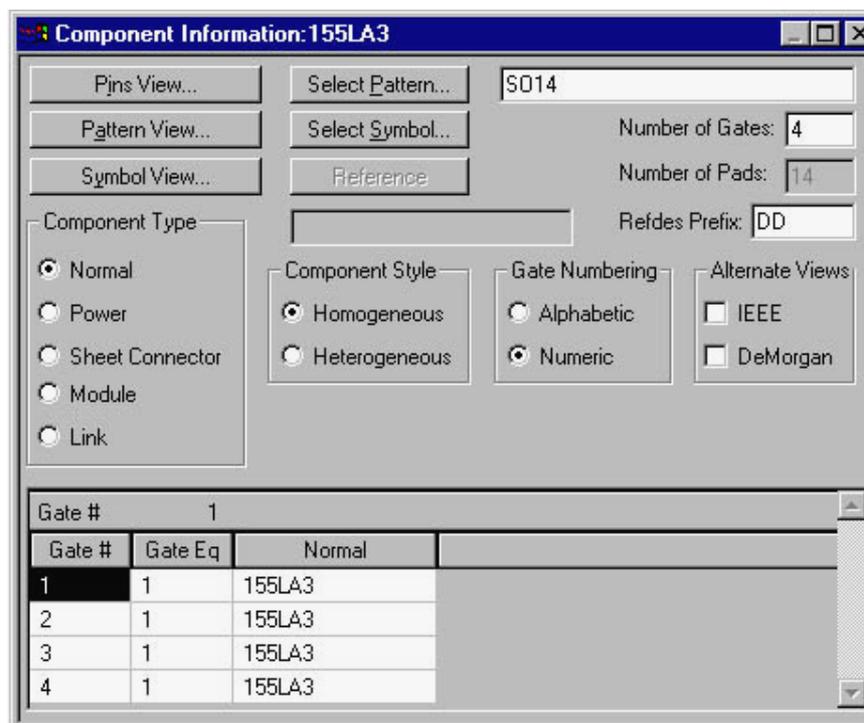


Рисунок 8.11 – Меню Component Information

1. В строке **Number of Gates** указывают число секций, в данном примере - 4. Число выводов компонента проставляется на строке **Number of Pads** автоматически. В строке **Refdes Prefix** указывают префикс позиционного обозначения компонента, в данном примере - DD. После этого становится доступной панель **Select Symbol**, после нажатия на которую выбирают имя основного изображения символа компонента, в данном примере это 155LA3.

2. В графе **Component Type** выбирают тип компонента *Normal*.

3. В графе **Component Style** для однородного компонента выбирают строку *Homogeneous*.

4. В графе **Gate Numbering** выбирают цифровой способ именования секций компонента *Numeric* (первая секция в графе **Gate #** в таблице на рис. 8.11 получит номер 1, вторая - 2 и т.п.). Все секции однородных компонентов по умолчанию получают одинаковый код логической эквивалентности **Gate Eq**, что позволит их автоматически переставлять в процессе размещения компонентов на ПП.

**Редактирование таблицы выводов компонента.** Создание компонента завершается заполнением таблицы выводов, которая выводится на экран нажатием кнопки **Pins View** (рисунок 8.12, а). Это самая трудоемкая часть работы по созданию компонента.

Во-первых, нужно установить соответствие между номерами выводов всех секций **Sym Pin #** и позиционными обозначениями выводов корпуса **Pin Des**.

В интегральной схеме серии K155 вывод 7 соединяется с «землей» (GND), а вывод 14 с источником питания (VCC).

Сначала скопируем содержание столбца **Pad #** в столбец **Pin Des**. Видно, что информация о номерах выводов секций 2, 3 и 4 не соответствует ИС серии K155,

и ее необходимо исправить, редактируя соответствующие строки таблицы.

Таблица выводов компонентов **Pins View** заполняется следующим образом. В нашем случае (см. рис. 8.12, а) мы видим, что во многих столбцах таблицы отсутствуют записи. Столбцы **Sym Pin #** и **Pin Name** заполнены только для одной 1-й секции, поэтому в столбце **Gate #** (туда заносятся номера секций) в строки 1, 2 и 3 необходимо записать единицы. Информация для остальных секций заполняется в соответствии с секцией 1, т.е. если для секции 1 известно, какие выводы символа сопоставить с реальными выводами микросхемы, то теперь необходимо выводы 2-, 3- и 4-й секции тоже логически связать с реальными выводами микросхемы, но на основе примера секции 1. Для этого нужно правильно заполнить столбцы **Sym Pin #** и **Pin Name**.

Pad #	Pin Des	Gate #	Sym Pin #	Pin Name	Gate Eq	Pin Eq	Elec. Type
1	1		1	in1			Unknown
2	2		2	in2			Unknown
3	3		3	out			Unknown
4	4						
5	5						
6	6						
7	7						
8	8						
9	9						
10	10						
11	11						
12	12						
13	13						
14	14						

а

Pad #	Pin Des	Gate #	Sym Pin #	Pin Name	Gate Eq	Pin Eq	Elec. Type
1	1	1	1	in1	1	1	Input
2	2	1	2	in2	1	1	Input
3	3	1	3	out	1		Output
4	4	2	1	in1	1	1	Input
5	5	2	2	in2	1	1	Input
6	6	2	3	out	1		Output
7	7	PWR		GND			Power
8	8	3	3	out	1		Output
9	9	3	1	in1	1	1	Input
10	10	3	2	in2	1	1	Input
11	11	4	3	out	1		Output
12	12	4	1	in1	1	1	Input
13	13	4	2	in2	1	1	Input
14	14	PWR		VCC			Power

б

Рисунок 8.12 – Таблица выводов компонента до (а) и после редактирования (б)

В соответствии с логической структурой компонента К155ЛА3 (см. рис. 8.9, а) ко второй секции относятся выводы 4, 5 и 6 микросхемы, поэтому в строки 4, 5, 6 столбца **Gate #** занесем двойки. Далее аналогичным образом перенесем значения столбцов **Sym Pin #** и **Pin Name** из секции 1, т.е. в строку 4 в столбцы **Sym Pin #** и **Pin Name** запишем значения «1» и «in1» соответственно, в строку 5 в эти же столбцы запишем значения «2» и «in2», в строку 6 запишем значения «3» и «out». Аналогичным образом заполняются остальные ячейки таблицы.

Вывод 7 должен получить имя **GND**, а вывод 14 - **VCC**, а в столбце - **Elec. Type** им необходимо присвоить тип **Power**. Полностью заполненная таблица представлена на рис. 8.12, б.

В столбце **Pin Name** можно вводить (если при создании символа компонента имя не задавалось) или изменять имена выводов. Для этого курсором выделим соответствующую ячейку и введем необходимую информацию, которая появляется в строке над таблицей. Нажатием кнопки  или **Enter** введенные данные переносятся в выделенную ячейку. Нажатие  или **Esc** отменяет ввод. Копирование данных выделенной ячейки (или нескольких ячеек) производят нажатием клавиши **Ctrl+C** или **Ctrl+Ins** (либо правой кнопки мыши, выбирая пункт в выпадающем меню), затем выделяют ячейку (или ячейки), куда нужно эти данные скопировать, и нажимают комбинацию клавиш **Ctrl+V** или **Shift+Ins** (обычная техника для Windows).

Эквивалентность секций (или выводов) подразумевает возможность менять местами секции (выводы) для удобства прокладки трасс автотрассировщиком печатных плат. В столбце **Gate Eq** указывается эквивалентность секций.

В нашем случае установка в любой ячейке любой секции единицы приведет к автоматической расстановке единиц и в ячейках, относящихся к другим секциям, потому что секции у нас одинаковые (т.е. компонент однородный), а значит, они эквивалентны.

Аналогичным образом необходимо заполнять и столбец **Pin Eq**. Для нашего случая необходимо занести единицы в строки с именами **in1** и **in2**. Эквивалентность выводов действует только в пределах одной секции.

В столбце **Elec. Type** указывается тип вывода, используемый при поиске ошибок в принципиальных электрических схемах. Нажатие на правую кнопку мыши (либо на кнопку  с выпадающим списком) открывает выпадающий список типов выводов (Electrical Type). Для быстрого выбора типа вывода достаточно напечатать первый символ его имени. Например, ввод символа **I** присваивает тип **Input**.

*Примечание.* Для простановки символа логической инверсии в имени вывода используется знак «~». Так, имя  $\overline{G}$  следует ввести как ~G, имя  $\overline{GC2A}$  - как GC~2A.

## Сохранение компонента в библиотеке

Перед сохранением в библиотеке компонента необходимо выполнить ко-

манду проверки **Component>Validate** (кнопка ). При обнаружении ошибок выводятся соответствующие информационные сообщения. Если же ошибок не обнаружено, программа сообщает: **No errors found!**

После исправления всех ошибок выполняется команда сохранения компонента в текущей библиотеке **Component>Save** или **Component>Save As**. Однако перед выполнением этих команд обязательно проверяется наличие ошибок, и пока ошибки не исправлены, компонент сохранить нельзя. Имя нового компонента указывают по дополнительному запросу (рисунок 8.13).

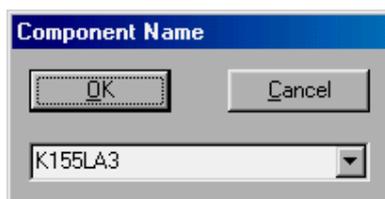


Рисунок 8.13. Сохранение компонента в текущей библиотеке

#### 8.4 Создание модели схемы электрической принципиальной в редакторе SHEMATIC P-CAD

Редактор PCAD SHEMATIC предназначен для построения схем электрических из библиотечных компонентов с последующей передачей информации об электрических связях между компонентами в PCAD PCB для проектирования печатной платы. PCAD SHEMATIC можно использовать также для оформления конструкторской документации (схем электрических принципиальных, перечней документов и т.п.).

Внешний вид экрана монитора при запуске PCAD Schematic показан на рисунке 8.14.

После запуска редактора схем PCAD Schematic рекомендуется настроить его конфигурацию: определить размер листа схемы и настроить другие параметры в меню команд **Options > Configure** (рисунок 8.15).

В графе **Units** выбирают систему единиц - миллиметры. Изменить систему единиц можно на любой фазе работы со схемой без потери точности. О текущей системе единиц можно судить по размерности ширины линий, указываемой на строке состояний.

Режим ввода цепей и линий устанавливаются в графе **Orthogonal Modes**: 90/90 Line-Line — ввод ортогональных линий, 45/90 Line-Line — ввод диагональных линий. При включении режима 90/90 Line-Line линии проводятся по осям координат или под произвольным углом, при включении режима 45/90 Line-Line — по диагоналям или под произвольным углом. Рекомендуется включить оба режима, тогда линии проводятся по осям координат, по диагоналям или под произвольным углом, что определяется дополнительным нажатием клавиши **O**.

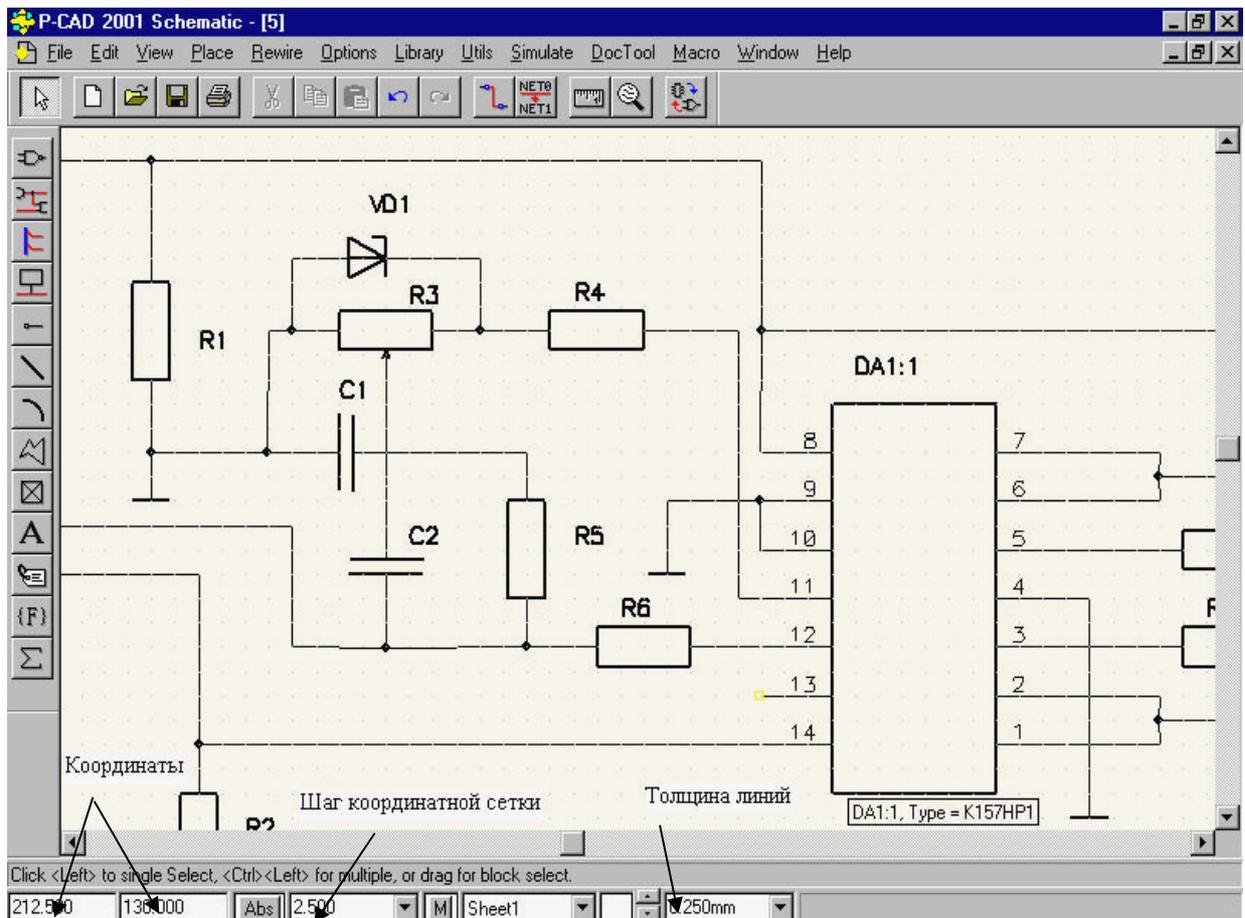


Рисунок 8.14 – Экран PCAD Schematic с загруженной схемой

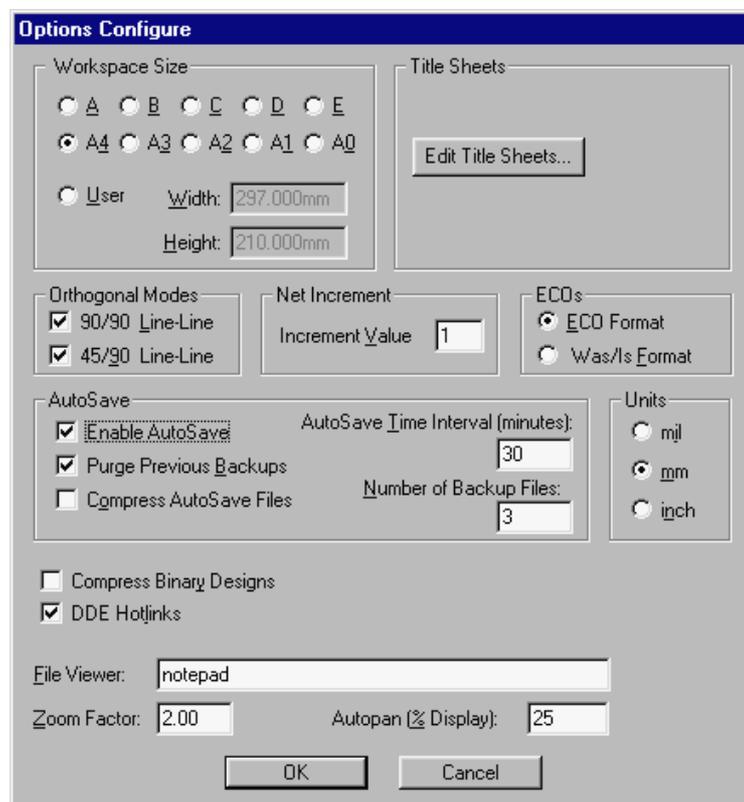


Рисунок 8.15 – Настройка конфигурации PCAD Schematic

Переключатель **DDE Hotlinks** устанавливает режим взаимного выделения цепей между графическими редакторами PCAD Schematic и PCAD PCB (так называемую горячую связь).

Шаг сетки устанавливают в меню **Option>Grids**. Текущее значение шага сетки выбирают курсором в списке Grids или, что более удобно, непосредственно на строке состояний.

В меню **Options>Display** задают цвета различных объектов, стиль изображения шин и ряд других параметров (рисунок 8.16).

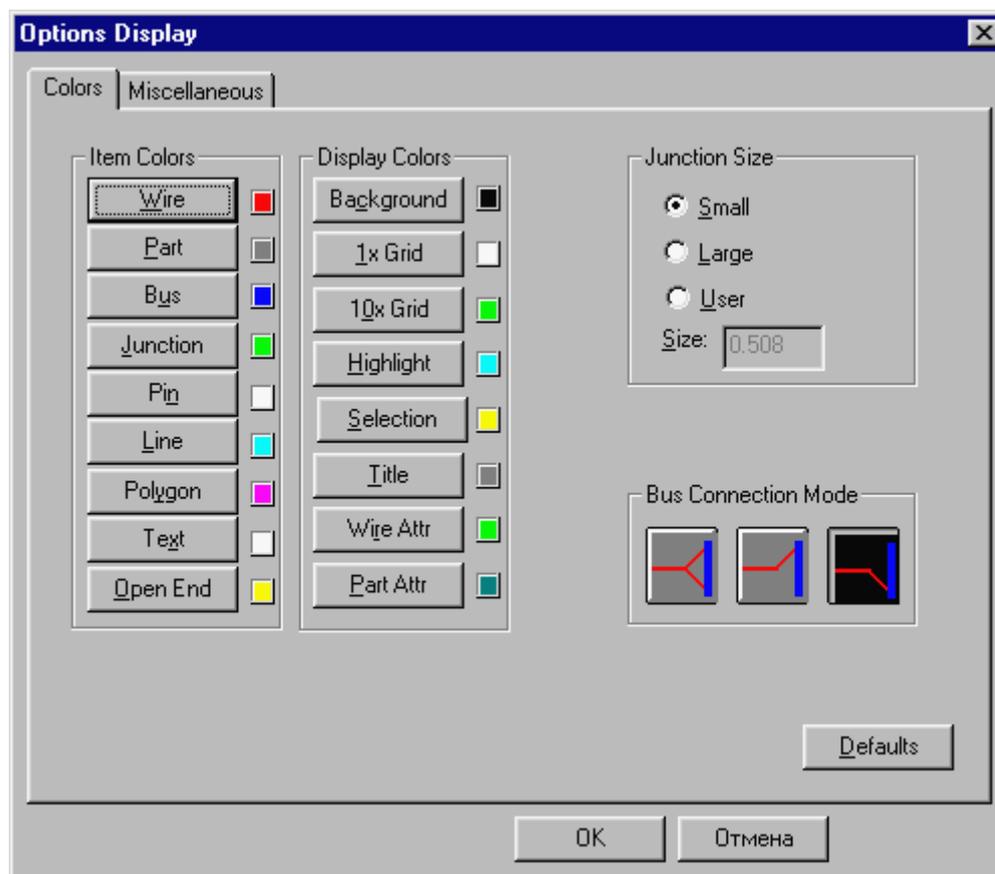


Рисунок 8.16 – Настройка параметров дисплея

В окне **Bus Connection Mode** указывают один из трех стилей изображения подсоединения цепей к шине.

В окне **Junction Size** выбирают размер точки электрического соединения цепей (точки «пайки»):

Small - малый (диаметр 20 mil); Large - большой (диаметр 30 mil); User - задаваемый пользователем (от 1 до 394 mil).

На закладке **Miscellaneous** на поле **ERC Errors** выбираются параметры:

Show - разрешение вывода на схему маркеров ошибок ERC; Hide - запрет вывода на схему маркеров ошибок ERC; No Change - запрет изменения на схеме маркеров ошибок ERC.

На панели **Cursor Style** выбирают тип курсора:

Arrow - стрелка; Small Cross - маленькое перекрестье; Large Cross - большое перекрестье.

На панели **Sheet Connector Cross Referencing** устанавливаются параметры соединителей страниц.

На панели **Miscellaneous** задают разнообразные параметры:

Draft Mode - изображение контуров линий и полигонов;

Display Default PinDes - отображение на экране номеров выводов символов компонентов, заданных по умолчанию;

Display Open Ends - отображение неподсоединенных выводов или цепей;

Display Overridden Errors - отображение на экране перекрывающихся маркеров ошибок;

Display Part Gate Number - отображение на экране номеров секций компонентов;

Scroll Bars - размещение на экране линий прокрутки;

Show Data Tips - вывод подсказок на рабочем поле (они не выводятся при использовании курсора большого размера и при выполнении команды **View>Snap to Grid**);

Thin Stroke Text - установка тонких линий векторных шрифтов;

Drag by Outline - изображение символов компонентов линиями контура при их перемещении или копировании (для ускорения перечерчивания экрана).

Нажатие клавиши **Defaults** назначает всем параметрам значения по умолчанию, **OK** - внесение изменений, **Cancel** - их отмена.

В PCAD Schematic нет понятия слоев изображения, которые можно сделать видимыми или невидимыми. Вся информация располагается на одном слое, и с помощью меню **Options > Display** можно любую информацию, например атрибуты компонентов (Part Attr), сделать невидимой — для этого ее нужно окрасить в цвет фона (Background).

По командам **Options > Current Wire** и **Options > Current Line** определяют ширину цепей Wire (осуществляют электрические соединения) и Line (графические линии не обеспечивают электрического соединения).

В меню **Options > Text Style** выбирают стиль текста, устанавливаемый по умолчанию, и при необходимости редактируют стили выполнения отдельных надписей. В меню редактирования шрифта выбирают тип шрифта: Stroke Font (векторные шрифты) или TrueType Font (контурные шрифты), имя шрифта (Font) и его размер. Все стили надписей сохраняются в файле схемы.

Параметры настройки конфигурации программы PCAD Schematic заносятся в файл SCH.INI и сохраняются при последующей работе с ней. Кроме того, параметры проекта (стили линий, стили шрифта и др.) заносятся в файл схемы. Поэтому целесообразно создать «шаблоны» пустых схем, например, в файлах template1.sch, template2.sch..., занося в них только параметры конфигурации (включая список шагов сетки, список стилей линии и т. п.), и загружать их перед началом создания новых схем.

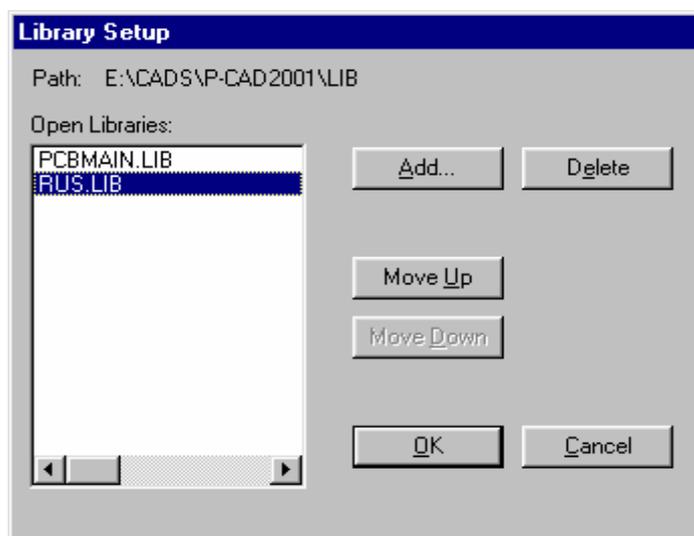
В PCAD имеется команда **File>Design Technology Parameters** для просмотра и загрузки набора технологических требований к печатной плате (ПП). Технологические правила, относящиеся к определению классов цепей и правил их трассировки и трассировки отдельных цепей, вводятся по команде **Op-**

tion>Design Rules как в редакторе PCAD Schematic, так и в редакторе PCAD PCB. Данные об апертурах, о наборе слоев ПП и стилях контактных площадок (КП) и переходных отверстий (ПО) можно модифицировать и ввести заново только в PCAD PCB.

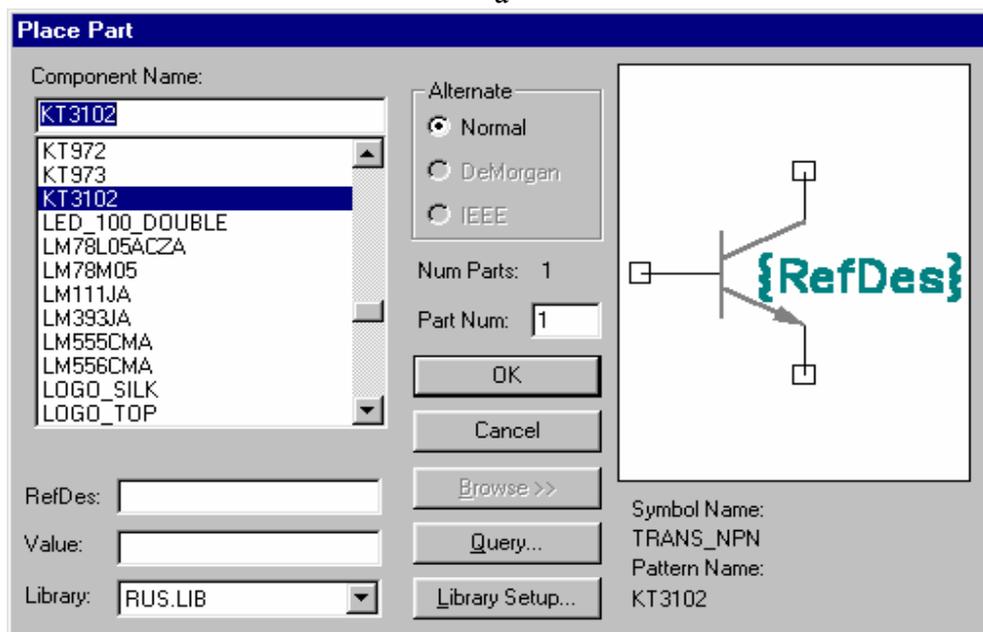
### Создание принципиальных схем

После настройки конфигурации PCAD Schematic приступают к созданию принципиальной схемы по следующему алгоритму.

**1. Загрузка библиотек.** Перед нанесением на схему символов компонентов по команде **Library>Setup** обеспечивается доступ к необходимым библиотекам (рисунок 8.17, а). Нажав клавишу **Add**, добавляют имена библиотек в список открытых библиотек (Open Libraries).



а



б

Рисунок 8.17 – Загрузка библиотек (а) и выбор символа компонента (б)

**2. Размещение компонентов на схеме.** В режим размещения символов компонентов на схеме переходят по команде **Place>Part** . После этого щелчок курсором в любой точке схемы открывает меню выбора компонента (рис. 8.17, б).

На панели **Library** указывается имя одной из открытых библиотек, список ее компонентов выводится в окне **Component Name**.

Имя нужного компонента выбирают из этого списка или вводят в верхней строке. Нажатие клавиши **Browse** позволяет просмотреть графическое изображение символа компонента (справа на рис. 8.17, б). При создании символа компонента в графе **Alternate** можно задать несколько альтернативных вариантов.

В графе **Num Parts** указывается общее число секций компонента (изменять их на этом этапе нельзя).

В окне **Part Num** по умолчанию указывается номер секции 1, но при необходимости его можно изменить перед размещением символа компонента на схеме.

Нажатие клавиши **Query** активизирует средства поиска компонента в открытых библиотеках. Сначала в диалоговом окне **Query** в столбцах **Criteria** задают критерии поиска и в столбце **Show** отмечают, какие характеристики найденных компонентов нужно отображать в таблицах результатов поиска.

Позиционные обозначения компонентов на схеме проставляются автоматически. Для изменения назначенного в библиотеке префикса позиционного обозначения его надо указать в поле **RefDes** в явном виде (см. рис. 8.17, б). Для компонентов, состоящих из одной секции, имя секции не проставляется.

При изображении многосекционных компонентов совмещенным способом (изображения смежных секций примыкают друг к другу) позиционные обозначения отдельных секций обычно не указываются, для этого необходимо параметры **RefDes** всех секций, кроме первой, сделать невидимыми (выключить кнопку **Visibility**), тогда будет видно позиционное обозначение только первой секции.

На строке **Value** проставляют номинал компонента, например сопротивление резистора.

После выбора в меню **Place>Part** нужного компонента и введения всех необходимых параметров нажимают **OK** — курсор примет форму перекрестья с разрывом в центре для точного позиционирования в узлах сетки. Непосредственное размещение символа компонента на схеме выполняется после щелчка курсором в любой точке рабочего окна. До тех пор, пока не отпущена левая кнопка мыши, символ перемещается по экрану. Он поворачивается на 90° в направлении против часовой стрелки и зеркально отображается нажатием клавиш **R** и **F** соответственно. Повторный щелчок курсором размещает на схеме очередную копию символа выбранного компонента, присваивая ему следующее позиционное обозначение, которое одновременно выводится в строке информации. Для увеличения на единицу номера секции компонента перед размещением символа нажимают клавишу **P**, а для увеличения позиционного обозначения

ния — клавишу **D**. Уменьшить эти значения можно одновременным нажатием клавиш **Shift+P** или **Shift+D**. Нажатие правой кнопки мыши или Esc прекращает ввод символов.

В режиме **Edit>Select** щелчок курсора в любой точке внутри контура символа компонента производит его выбор с возможностью перемещения, поворота и удаления символа вместе с принадлежащими ему атрибутами. Нажатие клавиши **Shift** и удерживание ее до щелчка курсора позволяет выбрать отдельный элемент символа: вывод, элемент графики, позиционное обозначение и др.

**3. Размещение электрических цепей.** После выбора команды **Place>Wire**  курсор принимает форму перекрестья. Щелчком мыши отмечается начальная точка цепи. Каждое нажатие левой кнопки мыши фиксирует точку излома. Нажатие клавиши **O** до отпускания левой кнопки мыши изменяет угол ввода линии из числа разрешенных (задаваемых в меню **Options>Configure**), нажатие клавиши **F** изменяет ее ориентацию. В связи с тем, что на схеме обычно преобладают горизонтальные или вертикальные цепи, в меню **Options>Configure** достаточно включить только режим ввода ортогональных цепей **90/90 Line-Line**. Завершается ввод цепи нажатием правой кнопки мыши (или Esc).

Ширина прокладываемой цепи (рекомендуется 0,25 мм) устанавливается по команде **Options>Current Wire: Thick** — широкая (0,381 мм = 15 mil), **Thin** — узкая (0,254 мм = 10 mil) и **User** - задаваемая пользователем (в пределах от 0,1 до 100 mil).

Неподсоединенные выводы компонентов и открытые концы цепей, не подсоединенных к другим цепям или к выводам компонентов, помечаются квадратами, которые гаснут после выполнения электрического соединения.

Если при перемещении цепи, имеющей присвоенное ей ранее имя (не системное имя типа **NETxxxxx**), она соприкоснулась с другой такой же цепью, открывается диалоговое окно (рисунок 8.18) для выбора имени объединенной цепи или отказа от выполнения операции. При одновременном соединении нескольких цепей (это возможно при их перемещении в окне) выводится сообщение об отказе от их объединения, что позволяет продолжить перемещение этих цепей или вернуться к исходному состоянию по команде **Undo**.



Рисунок 8.18 – Выбор имени соприкасающихся цепей

Включение в цепь дополнительной точки излома для последующего редактирования выполняется по команде **Rewire>Manual** .

В строке информации выводится присваиваемое автоматически имя размещаемой цепи, например Net: NET00003. Следующая цепь получит имя NET00004.

Присвоить цепи другое имя можно двумя способами.

I. Индивидуально выбрать каждую цепь и щелчком правой кнопки мыши открыть в меню редактирования пункт **Properties** и изменить имя цепи, заменив на строке **Net Name** назначенное системой имя другим, например заменить NET00001 на A1. Это имя будет *видимым*, если включить на закладке *Wire* переключатель **Display**.

II. С помощью команды **Utils>Rename Nets** в открывшемся меню на строке *Net Name* ввести префикс имени и выбрать параметр *Increment Name* (режим приращения имени). После закрытия этого меню щелчком курсора по первой цепи ей присваивается имя Pref 0, по второй - Pref 1 и т. д. Чтобы начать нумерацию цепей не с нуля, нужно начальное имя ввести в явном виде, например DATA5. Тогда первая цепь получит имя DATA5, вторая - DATA6 и т. д.

Если к цепи подсоединить компонент заземления GND, то она автоматически получит имя GND, так как выводу компонента заземления присвоен тип Power, что обеспечивает автоматическую замену имени цепи, к которому он подключен, на имя этого компонента.

Электрическое соединение пересекающихся цепей обозначается точкой «пайки» (**Junction**), которая автоматически проставляется на T-образных соединениях.

Чтобы проставить точку соединения пересекающихся цепей, нужно при вводе второй цепи щелкнуть курсором в точке пересечения и затем продолжить ее построение.

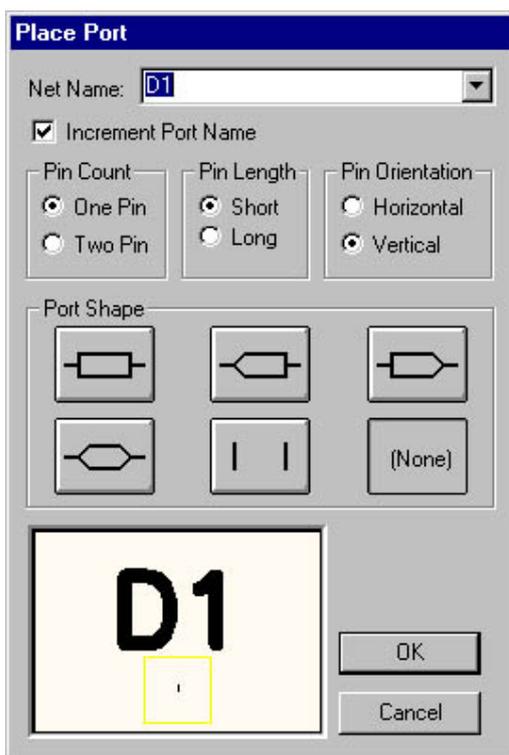
**4. Размещение шин.** Выбор команды **Place>Bus**  активизирует режим ввода шин. Щелчком курсора отмечают начало шины и точки ее излома. Построение шины завершается нажатием правой кнопки мыши или Esc.

После этого по команде **Place>Wire** размещают цепи. При рисовании цепи, входящей в состав шины, ее начало или конец должны располагаться в любой точке шины. При этом при подключении к шине автоматически изображается излом цепи под углом 45° (стиль этого изображения устанавливают в меню команды **Options>Display** (см. рис. 8.16, а). Шина изображается на схеме линией шириной 0,76 мм и изменить ее ширину нельзя. При необходимости можно поверх нее нарисовать линию нужной ширины по команде **Place>Line**.

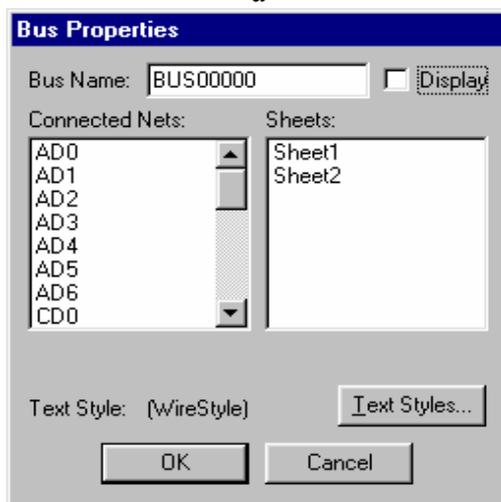
Имена цепей, образующих шину, задаются подключением к цепям по команде **Place>Port**  специальных портов. После выбора команды **Place>Port** щелчком мыши на любой точке схемы открывается изображенное на рисунке 8.19, а меню выбора порта. На расположенной сверху строке *Net Name* указывают имя первой цепи, например D1. Если включить переключатель *Increment Port Name*, то при размещении портов они автоматически получают имена

D1, D2, D3 и т. п.

Для обозначения цепей в шине в соответствии с ЕСКД необходимо выбрать тип порта без рамки (*None*) с одним выводом (*One pin*) и располагать его вертикально (*Vertical*). В результате имя цепи (порта) будет расположено над ней. После закрытия меню нажатием **ОК** порты подключаются последовательными щелчками мыши по именуемым цепям (текущее имя цепи отображается в строке информации). Именованные таким образом цепи являются глобальными (*Global*), их можно переименовывать по командам **Edit>Nets** и **Utils>Rename Nets**. Видимость имени шины устанавливается с помощью переключателя *Display* меню **Bus Properties** (рис. 8.19, б).



а



б

Рисунок 8.19 – Диалоговые окна команд Place>Port (а) и Bus Properties (б)

**5. Размещение проекта на нескольких листах.** Для создания дополнительных листов схемы по команде **Options>Sheets** на закладке *Sheets* в поле Sheet Name последовательно определяют имена второго и последующих листов схемы, каждый раз нажимая кнопку **Add**. Наиболее естественно именовать листы схемы Sheet1, Sheet2 и т. д. (при этом *все листы* находятся в *одном файле*). В дальнейшем листы схемы переключаются в строке состояний.

Перед переносом части схемы на другой лист необходимо к выводу компонента, связь от которого должна быть проложена на другой лист, подключить порт, имеющий один вывод (соединитель страниц), и присвоить ему имя цепи (рисунок 8.20). На другой странице должен быть размещен порт с тем же именем для обеспечения электрического соединения.

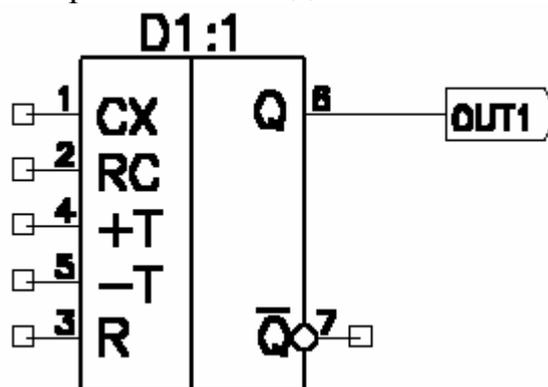


Рисунок 8.20 – Соединитель страниц

**6. Добавление атрибутов цепей и компонентов.** Для добавления атрибута цепи (или компонента) выбирается цепь (или компонент) и после открытия щелчком правой кнопки мыши меню редактирования указывают в нем пункт **Properties**. Затем по команде **Add** добавляют атрибуты, как показано на рисунке 8.21, где введен атрибут ширины проводника Width, который будет принят во внимание автотрассировщиком. С помощью выключателя **Visible** каждый атрибут может быть сделан видимым или невидимым на схеме. Для управления видимостью на экране всех однотипных атрибутов по команде **Option>Display** атрибуту можно назначить индивидуальный цвет или цвет фона (чтобы он был не виден). Аналогично при выводе схемы на печать по команде **File>Print>Print Options** указывают цвета атрибутов и других объектов. Таким образом назначают цепям или компонентам индивидуальные атрибуты. Атрибуты общего характера вводят по команде **Place>Attribute**.

**7. Поиск объектов.** Выберем компонент и щелчком правой кнопкой мыши развернем всплывающее меню, в котором укажем команду **Highlight Attached Nets** — в результате высвечиваются все цепи, подключенные к этому компоненту. Для поиска на схеме компонента выполняют команду **Edit>Parts** и в открывшемся меню со списком всех компонентов курсором указывают имя конкретного компонента. После нажатия на кнопку **Jump** этот компонент ярко высвечивается, даже если он находится на другом листе схемы (листы схемы переключаются автоматически). Аналогичную процедуру выполняют при выборе цепи в меню команды **Edit>Nets** (рисунок 8.22).

В списке цепей можно вывести имена всех цепей (All Nets) или только глобальных цепей (Global Nets Only), а также имена шин (Bus). По команде **Jump to Node** курсор указывает на схеме узел данной цепи, предварительно выбранный в списке Nodes, в котором приняты обозначения: U3-1 - это вывод 1 компонента U3.

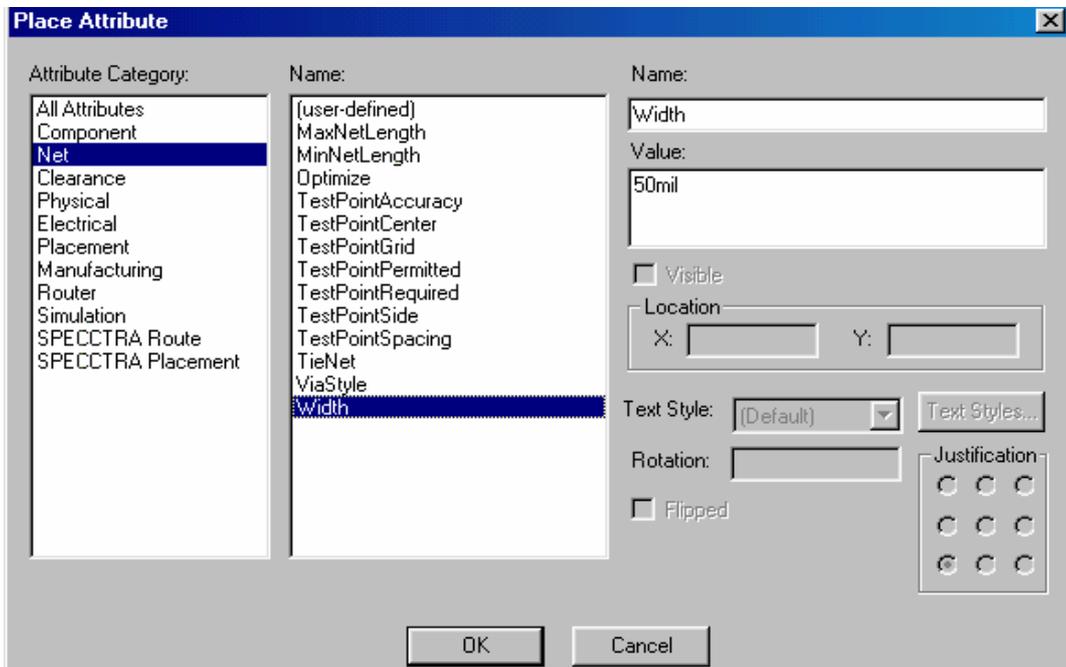


Рисунок 8.21 – Редактирование атрибутов цепей

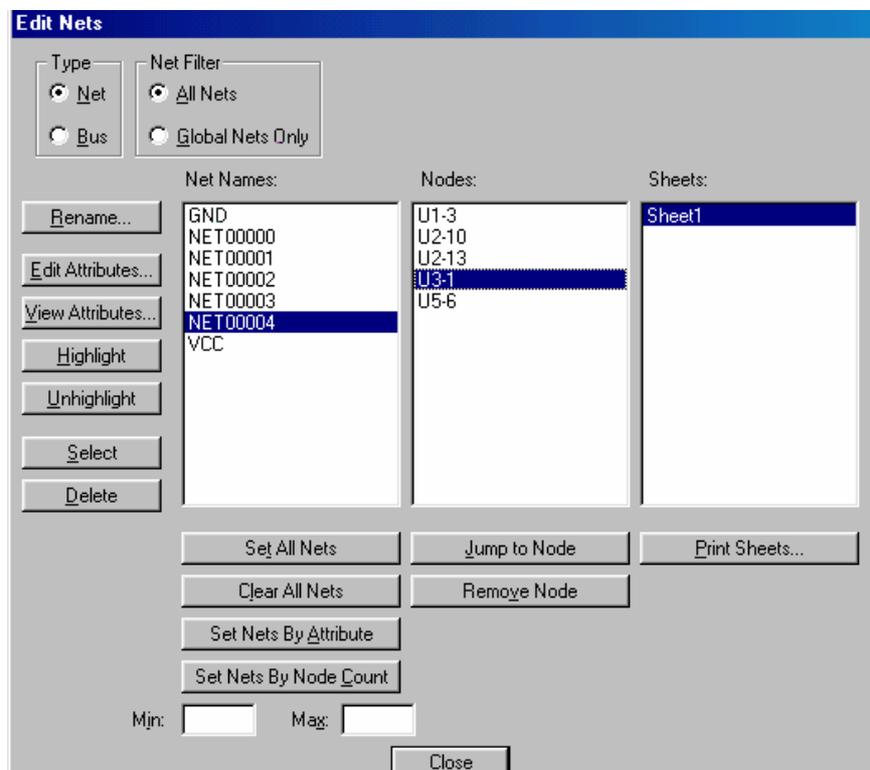


Рисунок 8.22 – Меню выбора цепей

**8. Редактирование схемы.** Для перемещения компонента или цепи необходимо выбрать их щелчком курсора и затем перемещать движением мыши.

Если после перемещения сегментов цепей часть из них будет расположена неаккуратно, щелчком курсора выбирают отдельный сегмент и перемещают его вершину. Выбор для редактирования элемента сложного объекта (например, позиционного обозначения компонента) производится щелчком левой клавиши мыши при одновременном нажатии кнопки **Shift**.

Перед перемещением группы объектов полезно щелчком правой кнопкой мыши выбрать команду **Select Point** для нанесения точки привязки (имеет форму перекрестья, заключенного в квадрат). При перемещении объектов в строке информации выводятся значения смещения точки привязки  $dX$ ,  $dY$  относительно ее первоначального расположения.

Скопировать выделенную группу объектов можно удерживанием клавиши **Alt** при перемещении мыши. При этом входящие в группу глобальные цепи не изменяют своих имен (к глобальным относятся цепи, к которым подключены порты или выводы компонентов, имеющие тип **Power**), остальные цепи переименовываются. Компонентам присваиваются новые позиционные обозначения. После размещения символов нескольких одинаковых компонентов всем им сразу можно присвоить одинаковые атрибуты.

**9. Изменение позиционных обозначений.** Перед завершением создания схемы целесообразно переименовать позиционные обозначения компонентов согласно ЕСКД вручную или с помощью специальной утилиты **DBX**. При перенумерации позиционных обозначений по команде **Utils>Renumber** невозможно соблюдать требования ЕСКД.

**10. Сохранение проекта.** По команде **File>Save** отредактированный проект сохраняется в исходном файле, по команде **File>Save As** проект сохраняется в другом файле, причем предварительно можно выбрать его формат:

*Binary Files* - бинарный формат (расширение имени \*.SCH);

*ASCII Files* - текстовый формат (расширение имени \*.SCH).

Файлы обоих форматов имеют одно и то же расширение имени \*.SCH. Бинарный формат более компактен и является основным, текстовый формат используют для обмена данными с другими программами.

В файле схемы хранятся следующие параметры: система единиц; набор шагов сетки; стили текста; шаблоны отчетов.

Чтобы не вводить эти данные всякий раз для каждой новой схемы, рекомендуется завести один или несколько отдельных файлов, не имеющих графической информации, ввести типичные параметры и сохранить эти шаблоны схем под уникальными именами, например **TEMPLATE\_1.SCH** и т.п.

**Иерархические структуры.** **PCAD Schematic** поддерживает иерархические или модульные структуры, что удобно для изображения принципиальных схем, состоящих из однотипных элементов, и для упрощения изображения сложных схем. На верхнем уровне иерархии помещается изображение модуля в виде прямоугольного «черного ящика», а его принципиальная схема размещается на более низком уровне иерархии (схемы всех уровней иерархии записываются на разных листах одного и того же проекта). Модуль представляется в виде компонента, имя которого (**Module name**) задает пользователь. Модуль

имеет выводы (Pin), к которым подключают внешние цепи. Связь между изображением модуля и его принципиальной схемой осуществляется с помощью понятия Link. Информация об этой связи (т. е. о схеме модуля) представляется в виде отдельного компонента, имя которого (Link name) также задает пользователь. Модуль иерархической структуры создают по команде **Utils>Module Wizard**. После размещения на схеме символов модулей необходимо выполнить команду **Utils>Resolve Hierarchy** для простановки сквозных позиционных обозначений компонентов и именования цепей проекта с учетом схем, входящих в него иерархических структур. Эта команда должна выполняться перед корректировкой схемы ЕСО и составлением списка электрических связей (поэтому, в частности, при «упаковке» схемы на ПП на нее переносятся и компоненты, входящие в состав иерархических структур).

### Проверка схем с помощью утилиты ERC

После создания принципиальной электрической схемы целесообразно выявить синтаксические ошибки, исправить их и только после этого перейти к разработке ПП. Проверку схемы выполняют по команде **Utils>ERC** (Electrical Rules Check). В основном меню этой команды задают перечень проверок, результаты которых приводятся в текстовом отчете:

Single Node Nets - цепи, имеющие единственный узел;

No Node Nets - цепи, не имеющие узлов;

Electrical Rules - электрические ошибки, когда соединяются выводы несовместимых типов (например, соединяются два выходных вывода или выходной вывод подключается к выводу источника питания);

Unconnected Pins - неподключенные выводы компонентов;

Unconnected Wires - неподключенные (плавающие) цепи;

Bus/Net Errors - входящие в состав шины цепи, встречающиеся только один раз (т. е. они входят в шины, но не выходят из них);

Component Rules - компоненты, расположенные поверх других компонентов;

Net Connectivity Rules - неправильное подключение цепей земли и питания;

Hierarchy Rules - ошибки иерархической структуры.

*Степень серьезности отдельных ошибок (Error, Warning или Ignored)* назначается в диалоговом окне, открываемом после нажатия на панель **Severity Level**. Там же необходимо включить опции *View Report* (Просмотр отчета сообщений об ошибках) и *Annotate Errors* (Индикация ошибок на схеме).

Сообщения об ошибках заносятся в файл *<имя схемы>.ERC*, изменение этого имени производят после выбора панели **Filename**.

Поиск ошибок в соответствии с заданной конфигурацией начинается после нажатия панели **OK**. Информация об ошибках помечается на схеме индикаторами и выводится в текстовый отчет.

Координаты ошибок приводятся в той системе единиц, которая выбрана в

меню команды **Options>Configure**.

### **Вывод данных**

Результаты проектирования выводятся в PCAD Schematic в виде:

- схемы, напечатанной на принтере или плоттере;
- списка соединений схемы (в частности, для упаковки схемы на ПП или выполнения моделирования);
- текстовых отчетов.

**Вывод схемы на печать.** Для подготовки к печати электрической схемы на принтере или плоттере выполняют команду **File>Print Setup** и в развернутом меню выбирают тип устройства вывода, предварительно установленного средствами **Windows**. Нажатием панели **Setup** открывают меню выбора размера и ориентации бумаги, характера передачи полутонов, разрешающей способности, качества печати (высокое, черновое). На строке **Minimum Line Width for printing (pixels)** устанавливают минимальную ширину прямых линий в пикселях (на дуги это не распространяется). При подготовке к печати указывается также масштабный фактор и поворот чертежа на  $90^{\circ}$ , если необходимо.

Непосредственный вывод на печать выбранных листов начинается после нажатия клавиши **Generate Printouts**, предварительный просмотр - после нажатия клавиши **Print Preview**.

**Создание списка соединений.** Список соединений включает в себя *список компонентов и цепей с указанием номеров выводов компонентов, к которым они подключены*. Он используется для так называемой процедуры «**упаковки схемы на печатную плату**» - размещения на поле ПП корпусов компонентов с указанием их электрических связей согласно принципиальной схеме. Для создания списка соединений необходимо по команде **Utils>Generate Netlist** открыть меню настройки параметров и в пункте меню **Netlist Format** выбирать формат списка соединений: *PCAD ASCII, Tango, FutureNet Netlist, FutureNet Pinlist, Master Designer, EDIF 2.0.0, PSpice, XSpice*.

Для разработки ПП с помощью графического редактора PCAD PCB рекомендуется выбирать формат **Tango** или **PCAD ASCII** (последний передает атрибуты схем на ПП). Имя файла списка соединений задают после нажатия клавиши **Netlist Filename**.

Нажатием кнопки **Include Library Information** включают в файл списка соединений (*только для формата PCAD ASCII*) информацию, необходимую для составления с помощью **Library Manager** библиотеки символов компонентов, находящихся в данной схеме (по команде **Library>Translate**).

**Составление отчетов.** По команде **File>Reports** создают текстовые отчеты о схеме. Дополнительно выбирается содержание и форма отчета, а также устройство вывода: консоль, принтер или файл. С помощью отчета удобно анализировать спроектированную схему и контролировать ошибки, внесенные оператором при создании библиотечных компонентов и вводе информации.

## 8.5 Проектирование печатной платы в PCB P-CAD

Для размещения компонентов на печатной плате и ручной трассировки в САПР PCAD используется графический редактор PCB.

Построен редактор печатных плат так же, как и редактор схем - отличие лишь в объектах, с которыми он работает. Поэтому на инструментальных панелях редактора имеются новые кнопки, а в меню - новые команды, необходимые для создания печатных плат. PCAD PCB в отличие от PCAD Schematic поддерживает многослойную организацию проекта.

### Настройка конфигурации P-CAD PCB

Перед началом работы в PCAD PCB необходимо настроить его конфигурацию по команде **Options>Configure**. Настройки PCAD PCB в основном совпадают с настройками редактора PCAD Schematic. Дополнительно в графе **Workspace Size** указывают размеры рабочей области, немного превышающие габаритные размеры ПП (по умолчанию устанавливается 254x254 мм, максимальные размеры ПП 60x60 дюймов или 1524x1524 мм).

На закладке *Route* отмечают опцию **T-Route by Default** для разрешения образования T-образных соединений проводников, в графе **Orthogonal Modes** включают все режимы (остальные параметры принимают значения по умолчанию, их настраивают позже по мере надобности).

На закладке *Manufacturing* задают значения глобальных технологических параметров: отступы масок пайки, зазоры между областью металлизации, направление пайки волной припоя и т.п.

При создании новой ПП по умолчанию устанавливаются 11 стандартных слоев:

**Top** - верхняя сторона ПП;

**Bottom** - нижняя сторона ПП;

**Board** - контур ПП;

**Top Mask** - маска пайки на верхней стороне ПП;

**Bot Mask** - маска пайки на нижней стороне ПП;

**Top Silk** - шелкография на верхней стороне ПП (контуры компонентов и т.п.);

**Bot Silk** - шелкография на нижней стороне ПП;

**Top Paste** - маска для паяльной пасты на верхней стороне ПП;

**Bot Paste** - маска для паяльной пасты на нижней стороне ПП;

**Top Assy** - вспомогательные данные на верхней стороне ПП;

**Bot Assy** - вспомогательные данные на нижней стороне ПП.

Слои Top и Bottom, Top Silk и Bot Silk и т.п. являются **парными**. Понятие парности слоев используется при переносе компонента на другую сторону ПП нажатием на клавишу **F** (**Flip** - зеркальное отображение), при этом вся графическая и текстовая информация переносится на ответствующие парные слои (при

зеркальном отображении простых графических объектов - линий, полигонов и т. п. - *они остаются на первоначальном слое*).

Всего в PCAD PCB может быть до **99** слоев. Слои создают и удаляют по команде **Options Layers**. Выделяют три типа слоев:

**Signal** - слои разводки проводников сигналов, помечают символом **S**;

**Plane** - слой металлизации для подключения цепей «земли» и «питания», помечают символом **P**;

**Non Signal** - вспомогательные слои, помечают символом **N**.

Каждый слой может быть включен (**Enable**, символ **E**) или выключен (**Disable**, символ **D**).

В графе **Routing Bias** указывают приоритетную ориентацию проводников на каждом слое при автоматической трассировке:

**Auto** - выбирается автоматически, символ **A**;

**Horizontal** - горизонтальная, символ **H**;

**Vertical** - вертикальная, символ **V**.

Удалять можно только слои, введенные пользователем, не являющиеся текущими и на которых не содержится информация; стандартные слои удалять нельзя.

Список значений ширины трасс проводников и геометрических линий составляется по команде **Options>Current Line**.

По команде **Options>Pad Style** открывают список стеков КП, по команде **Options>Vie Style** - список стеков ПО. Выбранные курсором в этих списках стеки являются текущими и помещаются на ПП при выполнении команд **Place>Pad**, **Place>Vie**. Нажатием на панель **Modify** открывают меню редактирования стеков КП (рисунок 8.23).

В PCAD PCB выделяют два вида стеков контактных площадок:

1) **простые (Simple)** - выводы *штыревых компонентов*, имеющие одинаковую форму КП на всех слоях, и *планарных компонентов*, имеющие КП только на одном слое;

2) **сложные (Complex)** - имеют различные КП на нескольких слоях.

Для стеков КП планарных компонентов задаются их геометрические размеры на том слое (Top или Bottom), на котором наносится графика корпуса (при последующей пайке планарных компонентов волной припоя их стеки КП необходимо размещать на слое Bottom).

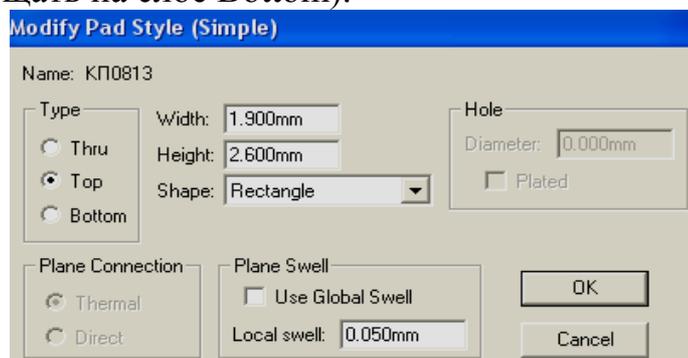


Рисунок 8.23 – Редактирование планарного стека

Перед размещением на ПП компонентов вручную или с помощью процедуры упаковки принципиальной схемы необходимо обеспечить доступ к библиотекам, в которых находятся эти компоненты. Библиотеки подключают по команде **Library>Setup**.

## Разработка печатных плат в PCAD PCB

Разработку новой ПП начинают с выполнения команды **File>New** и настройки конфигурации или загрузки шаблона. Затем на слое **Board** по команде **Place>Line** наносится *контур ПП* в виде замкнутой линейно-ломаной линии (дуги не разрешаются).

**1. Упаковка схемы на печатную плату.** При отсутствии принципиальной схемы проекта компоненты расставляют на ПП по команде **Place>Component** и по команде **Place>Connection** вводят электрические связи между их выводами. При наличии принципиальной схемы для переноса (упаковки) ее на ПП по команде **Utils>Load Netlist** загружают файл списка соединений (который создается в редакторе PCAD Schematic в форматах *Tango* или *PCAD ASCII*).

Обычно схема упаковывается на ПП, на которой *предварительно размещены*: разъемы; крепежные отверстия; другие компоненты, имеющие фиксированное положение (в диалоговом окне **Properties** этих компонентов необходимо отметить опцию **Fixed**), и *проложен* ряд трасс.

В этом случае после загрузки команды **Utils>Load Netlist** выводится сообщение, информирующее о необходимости соблюдения следующих ограничений:

— компоненты с совпадающими на ПП и на схеме позиционными обозначениями (*RefDes*) должны иметь одинаковые типы корпуса (*Type*). При обнаружении конфликтов упаковка не производится;

— все присутствующие на ПП компоненты, не входящие в список соединений, будут сохранены;

— на ПП переносятся все компоненты из списка соединений, которые не установлены на ней предварительно;

— электрические связи, проложенные предварительно на ПП и отсутствующие в списке соединений, удаляются (обновляется вся информация об электрических связях), однако все проложенные ранее проводники сохраняются, даже если они отсутствуют в списке соединений;

— после выполнения команды нельзя восстановить первоначальный вид ПП с предварительно размещенными компонентами, поэтому ее рекомендуется сохранить в отдельном файле.

**2. Размещение компонентов на плате.** После загрузки списка соединений (упаковки схемы) на ПП приступают к размещению компонентов внутри контура ПП (если он есть). Размещение компонентов на ПП обычно производят вручную. Линии электрических связей, перемещаемые вместе с компонентами, помогают правильно их разместить.

Нажатием клавиши **R** поворачиваем выбранный объект против часовой

стрелки на 90°, одновременным нажатием **Shift+R** поворачиваем его на угол, заданный в меню **Options>Configure** на закладке *General* (параметр **Rotation Increment**), нажатием клавиши **F** зеркально отображаем объекты относительно оси Y и переносим компоненты на противоположную сторону ПП.

При размещении однотипных компонентов удобно их автоматически выравнивать. Для этого выравниваемые компоненты по очереди выбираются щелчком курсора (при выборе второго и последующих компонентов нажимают и удерживают клавишу *Ctrl*). Затем выбирают опцию **Align** (Выравнивание) и в открывшемся меню отмечают способ выравнивания.

После завершения размещения компонентов полезно выполнить минимизацию длин соединений путем перестановки логически эквивалентных секций компонентов и их выводов по команде **Utils>Optimize Nets** (перестановка возможна, если в ней не участвуют уже проложенные проводники).

**3. Задание правил проектирования.** Перед началом трассировки в меню **Options>Grids** задают необходимый шаг сетки.

Затем по команде **Options>Design Rules** на закладке *Layers* устанавливают **допустимые зазоры** для каждого слоя трассировки (рисунок 8.24).



Рисунок 8.24 – Пример задания правил и ограничений трассировки

Перед началом ручной трассировки целесообразно включить режим текущей проверки допустимых зазоров, пометив на закладке *Online DRC* команды **Options>Configure** строку **Enable Online DRC**.

Для управления размещением компонентов и трассировки соединений в автоматическом или интерактивном режиме и проверки DRC цепям и компонентам присваивают атрибуты по команде **Edit>Nets**.

**4. Барьеры трассировки.** Графический редактор PCAD PCB имеет возможность создавать на ПП барьеры трассировки. Расположение на ПП по команде **Place>Keepout** сегмента линии или замкнутой области в виде полигона запрещает Quick Route пересекать их трассами проводников на одном выбранном или на всех доступных слоях (места размещения крепежных отверстий, запрещенные для трассировки по конструкционным соображениям области и т.п.). Предварительно по команде **Options>Current Keepout** устанавливают параметров барьеров трассировки: линия (Line) или полигон (Polygon), располагаемые на текущем (Current) или на всех доступных слоях (All).

Контур ПП, рисуемый на слое Board по команде **Place>Line**, воспринимается Quick Route как барьер, который нельзя пересекать проводниками на всех слоях. Поэтому трассы будут проведены как внутри этого контура, так и вне его (но не пересекая).

**5. Ручная трассировка соединений.** Перед началом трассировки какого-нибудь проводника задают его ширину и выбирают нужный слой. Трассу проводника можно начать только от тех КП, которые имеют электрические связи. При необходимости электрические связи устанавливают вручную по команде **Place>Connection**.

Ручную трассировку проводников выполняют по команде **Route>Manual**.

Для облегчения создания T-образного соединения трасс выполняют его при нажатой клавише **Shift** или заранее на закладке *Route* диалогового окна команды **Options>Configure** включают опцию **T-Route by Default**.

При смене в процессе прокладки трассы текущего слоя нажатием клавиш **L**, **Shift+L** или с помощью строки состояний трасса продолжается на другом слое, при этом автоматически вставляется ПО текущего типа (заданного заранее по команде **Options>Via Style**).

Не прерывая прокладку трассы, можно изменить ширину проводника по команде **Options>Current Line** или с помощью строки состояний.

Нарушение зазора между трассой проводника и выводом компонента, ПО или трассой другой цепи отмечается индикаторами ошибок в виде круга с перекрестием, если включен режим **Online DRC**.

Нажатие правой кнопки мыши или выбор любой команды завершает трассировку проводника по кратчайшему расстоянию до контактной площадки вывода компонента. Нажатием косой черты «/» или «\» прекращают прокладку трассы без ее завершения.

Завершение трассы точно на контактной площадке пункта назначения обозначается ромбом с перекрестьем.

В процессе трассировки полезно обращать внимание на строку информации, в которой выводятся следующие данные:

- приращения координат dX, dY каждого сегмента, пока нажата левая кнопка мыши;
- общая длина трассы до точки излома **Total length** (с учетом длин дуг);
- число ошибок при прокладке трассы «<n> error(s) during routing» (выводится после ее завершения).

**6. Интерактивная трассировка соединений.** Трассу прокладывают по команде **Route>Interactive** движением курсора при нажатой левой кнопке, при этом автоматически огибаются препятствия (проводники, КП, ПО и области металлизации), соблюдаются допустимые зазоры. Нажатием правой кнопки мыши в процессе прокладки трассы открывают следующее меню:

**Complete** - завершение прокладки трассы (если это возможно) с соблюдением установленного ранее режима ввода проводников (ортогонально или по диагонали) и допустимых зазоров;

**Push Traces** - включение режима отталкивания мешающих проводников;

**Suspend** - прекращение прокладки трассы с сохранением проложенного участка;

**Cansel** - прекращение прокладки трассы с отменой ввода последнего сегмента (аналог нажатия клавиши *Esc*);

**Options** — изменение параметров конфигурации проекта по команде **Options>Configure**;

**Layers** — открытие окна команды **Options>Layers** для изменения структуры слоев платы;

**Via Style** — открытие окна **Options>Via Style** для выбора типа ПО или его редактирования;

**Unwind** — отмена прокладки последнего сегмента проводника (аналог нажатия клавиши **Backspace**).

Клавиши **O**, **F**, «/», «\» и стрелки имеют такие же назначения, как и при ручной прокладке трасс. Единственное отличие — не производится скругление трасс по дуге.

Когда в процессе прокладки трассы левую кнопку мыши отпускают в точке окончания линии электрической связи, прокладка трассы завершается и можно переходить к прокладке следующей.

**7. Сглаживание прямоугольных изгибов проводников.** Прямоугольные изгибы проводников скашиваются под углом 45° или сглаживаются дугами по команде **Route>Miter**. Режим сглаживания устанавливают на панели **Miter Mode** закладки *Route* команды **Options>Configure**. Сглаживание начинают щелчком курсора в точке излома трассы, не отпуская левую клавишу мыши, перемещают курсор и устанавливают необходимые размеры изгиба трассы.

По завершении ручной трассировки проводников полезно выполнить команду **Utils>Trace Clean-up** для удаления наложенных друг на друга сегментов трасс и лишних точек излома.

**8. Области металлизации.** На слоях сигналов могут располагаться области металлизации, электрически подсоединяемые к одной из цепей и автоматически отделяемые зазорами от других цепей и КП. Эти области создают в два этапа:

1) по команде **Place>Copper Pour** рисуют внешний контур области металлизации в виде полигона (пересечения сторон полигона не допускаются);

2) область выбирают щелчком курсора, в выпадающем меню выбирают строку **Properties** и затем на закладке *Connectivity* открывшегося меню **Copper Pour Properties** указывают имя цепи, к которой она должна быть подключена. Здесь же выбирают необходимость использования КП с тепловыми барьерами (**Thennals**) или непосредственного соединения (**Direct Connection**) и задают ширину теплового барьера. После этого на закладке *Style* указывают особенности выполнения металлизации.

При прокладке проводников через область металлизации зазоры образуются автоматически, если на закладке *General* команды **Options>Configure** включена опция **Auto Plow Copper Pours**.

**9. Сохранение проекта.** По команде **File>Save** либо **File>Save As** отредак-

тированный проект сохраняется в файле, причем предварительно можно выбрать его формат.

**10. Проверка печатной платы с помощью утилиты DRC.** Перед завершением разработки ПП и выпуском фотошаблонов необходимо по команде **Utils>DRC (Design Rule Checking)** проверить ПП на соответствие принципиальной схеме и проверить соблюдение технологических ограничений. В меню этой команды выбирают различные правила проверок.

После выполнения проверок перечень ошибок заносится в файл с расширением имени \*.DRC. Включение опции **Annotate Errors** помечает на ПП места ошибок специальными индикаторами. После исправления ошибки ее индикатор автоматически удаляется.

### **Вывод данных**

Результаты разработки ПП выводят на принтеры и плоттеры различных типов, используя средства Windows. В меню команды **File>Print** в поле **Minimum Line Width (pixels)** устанавливают минимальную ширину линий, затем нажимают кнопку **Setup Print Jobs** и переходят в меню составления заданий. В этом меню сначала каждому заданию в графе **Print Name** присваивают имя. Затем в списке слоев **Layers** указывают нужные (второй и последующие слои выбирают щелчком курсора при нажатой кнопке *Ctrl*).

В заключение устанавливают опции печати:

**Scale** — масштаб изображения;

**X и Y offset** — смещение изображения по горизонтали и вертикали от края бумаги;

**Drill Symbols Size** — размер символов отверстий;

**Rotate** — поворот изображения на 90° по часовой стрелке;

**Mirror** — зеркальное отображение;

**Draft** — вывод контуров линий;

**Thin Striked Text** — изображение векторных шрифтов тонкими линиями;

**RefDes, Type, Value, Pads, Vias, Pad/Via Holes, Pick and Place, Glue Dot, Test Point, Keepout, Cutout, Connections, Mt Hole, Titles** - печать соответствующих объектов.

Нажатие на кнопку **Print Preview** позволяет просмотреть изображение, печать производится после нажатия на кнопку **Generate Printouts**.

**Создание и просмотр Gerber-файлов.** PCAD PCB обеспечивает создание управляющих файлов в стандартном формате фотоплоттера Gerber по команде **File>Export>Gerber** и загрузку изображения фотошаблона для контроля ошибок по команде **File>Import>Gerber**.

**Вывод информации для сверления отверстий.** По команде **File>Export>N/C Drill** из базы данных спроектированной ПП извлекается информация о координатах отверстий и создается текстовый управляющий файл в формате станка с ЧПУ типа Excellon (аббревиатура N/C означает Numerically Controlled — числовое программное управление (ЧПУ)).

**Составление списков соединений.** Список соединений включает в себя список компонентов и цепей с указанием номеров выводов компонентов, к которым они подключены. Изолированные области металлизации, а также неподсоединенные выводы компонентов в этот список не заносятся.

По команде **Utils>Generate Netlist** открывается меню настройки параметров. В пункте меню **Netlist Format** выбирают формат списка соединений: PCAD ASCII или Tango. Нажатием кнопки **Include Library Information** включают в файл списка соединений (только для формата P-CAD ASCII) информацию, необходимую для составления с помощью Library библиотеки корпусов компонентов, находящихся в данном проекте (по команде **Library>Translate**).

По команде **File>Reports** создают текстовые отчеты о ПП.

## 8.6 Автоматическая трассировка проводников

### Программа автоматической трассировки Quick Route

Программа автотрассировки Quick Route (файл QROUTE.EXE) поставляется совместно с PCAD PCB. Вызывают Quick Route из управляющей оболочки PCAD PCB по команде **Route>Autorouters** (рисунок 8.25) из списка:

Quick Route – трассировщик Quick Route;

PCAD Shape Route – трассировщик бессеточного типа, использующий алгоритмы оптимизации нейронных структур;

SPECCTRA – программа SPECCTRA в режиме трассировки.

Из этих программ Quick Route является наиболее простым трассировщиком, он пригоден для быстрой разработки не очень сложных ПП.

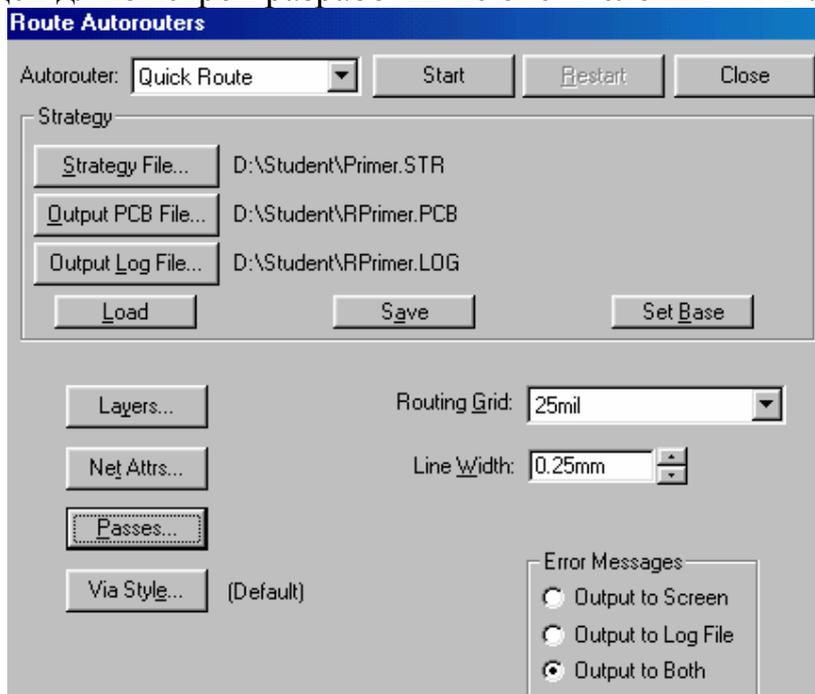


Рисунок 8.25 – Выбор автотрассировщика Quick Route

**Подготовка к трассировке.** Quick Route трассирует текущую ПП, загру-

женную в PCAD PCB. На ней должны быть размещены все компоненты и указаны электрические связи между выводами. Предварительно на ПП можно разместить барьеры трассировки Кеероут и некоторые проводники. Quick Route не изменяет предварительно размещенные проводники и не прокладывает трассы заново по более короткому пути.

Характер меню настройки стратегии трассировки зависит от выбранного типа трассировщика.

В меню Quick Route (см. рис. 8.25) в нижней части экрана расположены следующие кнопки:

**Layers** – конфигурация слоев. Quick Route поддерживает до четырех слоев металлизации;

**Net Attrs** – редактирование атрибутов цепей;

**Passes** – выбор проходов трассировки;

**Via Style** – выбор типа ПО.

На строке **Line Width** выбирают ширину проводника от 0,1 mil (0,01 мм) до некоторого значения, зависящего от выбранного шага сетки. Например, для шага сетки 25 mil оно составляет 12 mil в английской системе или 0,305 мм в метрической системе, не более. Ширину индивидуального проводника назначают с помощью атрибута WIDTH, который может принимать любое значение.

В меню редактирования стратегии трассировки **Pass Selection** выбирают типы проходов трассировки, выполняемых в следующем порядке:

**Wide Line Routing** – разводка всех широких цепей, имеющих атрибуты ALITOROUTEWIDE и WIDTH перед выполнением других проходов. На этом проходе прокладывают только горизонтальные и вертикальные трассы. Наклонные широкие трассы приходится прокладывать предварительно вручную – Quick Route их ширину не изменит;

**Horizontal** – выполнение простейших соединений по горизонтали на любом слое без использования ПО и с минимальным отклонением от прямых линий;

**Vertical** – выполнение простейших соединений по вертикали на любом слое без использования ПО и с минимальным отклонением от прямых линий;

**'L' Routes (1 via)** – формирование пересечения двух проводников и одного ПО, имеющего форму буквы L. Проводники располагаются на двух активных слоях и имеют противоположную ориентацию (горизонтальную или вертикальную);

**'Z' Routes (2 vias)** – формирование пересечения трех проводников и двух ПО, имеющего форму буквы Z. Проводники располагаются на двух активных слоях и имеют противоположную ориентацию (горизонтальную или вертикальную);

**'C' Routes (2 vias)** – формирование пересечения трех проводников и двух ПО, имеющего форму буквы C. Проводники располагаются на двух активных слоях и имеют противоположную ориентацию (горизонтальную или вертикальную). Трассировка типа C более гибкая, чем трассировки типа L и Z, так как проводникам разрешается размещаться на расстоянии более 100 mil вне прямо-

угольника, вершины которого находятся в соединяемых выводах;

**Any Node (2 vias)** – попытка провести проводники между любыми узлами цепи для обеспечения наиболее полной разводки при простановке не более двух ПО (на предыдущих проходах проводники разводились из условия минимизации их длины);

**Maze Routes** – трассировка типа «лабиринт», способная найти путь для оптимальной прокладки проводника, если это физически возможно. Максимальное число ПО в каждой цепи назначают с помощью атрибута MAXVIAS, который по умолчанию равен 10. Если лабиринтная трассировка заблокирует разводку ряда цепей, то поступают следующим образом: 1) выключают алгоритм Maze и с помощью Quick Route разводят ПП; 2) разводят ряд проводников вручную с помощью PCAD PCB; 3) завершают трассировку с помощью Quick Route, включив алгоритм Maze;

**Any Node (maze)** – попытка провести проводники между любыми узлами цепи с помощью алгоритма «лабиринт» (необязательно оптимальным образом);

**Route Cleanup** – улучшение внешнего вида ПП (manufacturing). На этом проходе часть проводников разводятся заново для их спрямления, где это возможно;

**Via Minimization** – минимизация количества ПО. Если необходимо сохранить расположение ряда предварительно размещенных ПО, следует или не использовать данный проход, или заменить эти ПО выводами со штыревыми отверстиями.

Трассировка начинается после нажатия на клавишу **Start** в меню Quick Route. При этом одновременно изменяется вид экрана, как показано на рисунке 8.26.

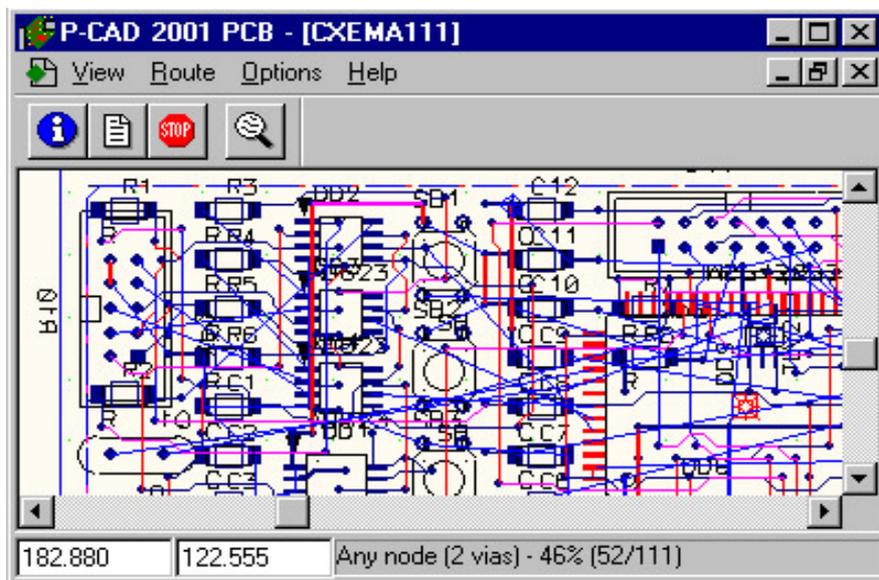


Рисунок 8.26 – Экран трассировщика Quick Route

По команде **Route>Info** выводится текущая информация о результатах трассировки. По команде **Route>Cancel** трассировка прекращается.

В рабочем окне на экране приводится изображение ПП. После прокладки проводника его изображение сразу же появляется на экране. Сообщение **Route**

**completed** указывает о завершении трассировки. Одновременно выводятся данные о проценте разведенных цепей. Кроме того, в скобках приведено число разведенных цепей и общее число цепей, отделенное косой чертой.

Информация о стратегии трассировки, результатах выполнения отдельных фаз трассировки и итоговые данные помещаются в файл протокола, имеющий расширение имени .LOG.

#### **Ограничения Quick Route:**

- разрешены только простые КП и ПО (имеют одну и ту же форму на всех слоях), глухие межслойные ПО не допускаются;
- для цепей, не имеющих атрибута AUTOROUTEWIDE, допускается только один стиль ПО;
- диаметр ПО не может более чем в два раза превышать текущий шаг сетки трассировки;
- разрешенные размеры сетки трассировки: 12,5 mil, 16,7–16,6–16,7 mil, 20 mil и 25 mil. Метрическая сетка не разрешается;
- ширина проводника не может быть больше половины шага сетки;
- для ПО нельзя создать специальную сетку;
- выводы компонентов можно повернуть только на 90°;
- допускается не более 4 слоев металлизации.

**Расположение объектов вне сетки трассировки.** Если вывод компонента не совпадает с узлом выбранной сетки, то трасса проводника подводится к ближайшему пути и от него будет проложен короткий сегмент до центра КП. Поэтому не совпадающие с узлами сетки выводы компонентов способствуют блокированию каналов трассировки. Поэтому рекомендуется так располагать компоненты, чтобы все их выводы совпадали с узлами сетки.

**Особенности трассировки компонентов с планарными выводами.** Quick Route позволяет выбрать предпочтительную ориентацию проводников на каждом слое трассировки. Для двусторонних ПП обычно выбирается горизонтальная ориентация проводников на верхнем слое (Top) и вертикальная на нижнем (Bottom). Однако это соглашение препятствует трассировке компонентов с планарными выводами. Например, если на верхней стороне ПП размещен планарный разъем, ориентированный параллельно нижнему краю ПП, к его средним выводам будет очень трудно подвести проводники. Для облегчения автоматической разводки компонентов с планарными выводами Quick Route генерирует рядом с каждым планарным выводом сквозное ПО (стрингер) и соединяет его с ним коротким проводником. После завершения разводки неиспользованные ПО автоматически удаляются.

По умолчанию ПО имеют на внутреннем слое металлизации КП с тепловыми барьерами. Если же двусторонняя ПП имеет металлизированный нижний слой, то на нем применяется непосредственное соединение контактных площадок (Direct Connect).

**Выбор сетки трассировки.** В Quick Route можно выбрать одну из четырех сеток трассировки: регулярные сетки с шагом 25, 20, 12,5 mil и нерегулярная сетка 16,7-16,6–16,7 mil. Перед выбором шага сетки и ширины проводников не-

обходимо выяснить на предприятии, где будет изготавливаться ПП, принятые технологические нормы.

Выполнение предварительных расчетов по выбору сетки и расположение компонентов абсолютно необходимо. Если разместить компоненты в сетке с шагом 50 mil, то возможен выбор шага сетки трассировки 25, 16,7-16,6–16,7 или 12,5 mil. Наименьший шаг сетки размещения, при которой возможно применение всех имеющихся в Quick Route сеток разводки, составляет 100 mil. При несогласованном выбор сеток размещения и разводки многие выводы компонентов не будут попадать в узлы сетки разводки, что не позволит достичь наиболее высокой степени разводки.

**Выбор размеров контактных площадок.** Кроме выбора оптимального шага сетки размещения компонентов необходимо выбрать размеры КП, чтобы обеспечить возможность прокладки нескольких проводников между соседними выводами. В табл. 3.1 приведены рекомендуемые значения максимальных размеров КП и ПО для каждого шага сетки трассировки.

Таблица 8.1 – Рекомендуемые размеры контактных площадок и ПО

Шаг сетки, mil	Рекомендуемый диаметр, mil		Ширина планарных выводов, mil
	КП штыревых выводов	ПО	
25	62	40	–
20	50	40	–
16,7-16,6-16,7	40	40	–
12,5	54	32	30

### **Бессеточный трассировщик Shape-Based Router**

В программе Shape-Based Router используются принципы оптимизации нейронных сетей, и она в основном предназначена для трассировки многослойных ПП с высокой плотностью размещения компонентов в автоматическом, интерактивном и ручном режимах; особые преимущества имеет при наличии планарных компонентов, выполненных в разных системах единиц (метрической и английской).

Кроме того, эта программа основана на так называемой Shape-Based, или *бессеточной* технологии. Согласно ей все объекты ПП моделируются в виде совокупности геометрических фигур (прямоугольник, круг, дуга, трасса, полигон). В отличие от привязанных к сеткам технологиям (Grid-Based), используемым, в частности, в PCAD Quick Route, при бессеточной технологии каждый объект моделируется не набором узлов сетки, а геометрически точно, за счет чего достигается более плотный монтаж. Характерная особенность бессеточной технологии – меньшие затраты памяти компьютера.

Программа Shape-Based Router нацелена на решение глобальных проблем

(100 %-ная трассировка, минимизация количества ПО и др.) и не позволяет выполнить тонкую настройку стратегии трассировки (например, определенную цепь заключить в экран, провести дифференциальную пару проводников, проложить индивидуальную цепь заданной длины и т. п.).

Shape-Based Router обрабатывает ПП, имеющие до 30 слоев, до 4000 компонентов, до 5000 выводов в одном компоненте, до 10 000 цепей и до 16 000 электрических соединений.

Программа Shape-Based Router может быть вызвана автономно из среды Windows (файл SR.EXE) или из меню **Route PCAD PCB** (команда **Autorouters**). В диалоговом окне вызова этой программы можно только указать имена файлов: исходной ПП в формате PCAD Route File (\*.PRF), выходной (оттрассированной) ПП в формате PCAD PCB (\*.pcb), протокол команд (\*.LOG). По умолчанию эти файлы получают те же имена, что и файл исходного проекта, но в начале имени выходных файлов добавляется префикс R.

В Shape-Based Router передаются из PCAD PCB значения ширины трасс, присвоенные цепям с помощью атрибутов Width, и тип ПО, установленный по умолчанию. В качестве глобального зазора Primary Clearance принимается зазор Line to Line, заданный для слоя Bottom (его можно в дальнейшем изменить). Трассы, проложенные вручную, защищаются от изменений по команде **Edit>Fix**. Экран Shape-Based Router изображен на рисунке 8.27.

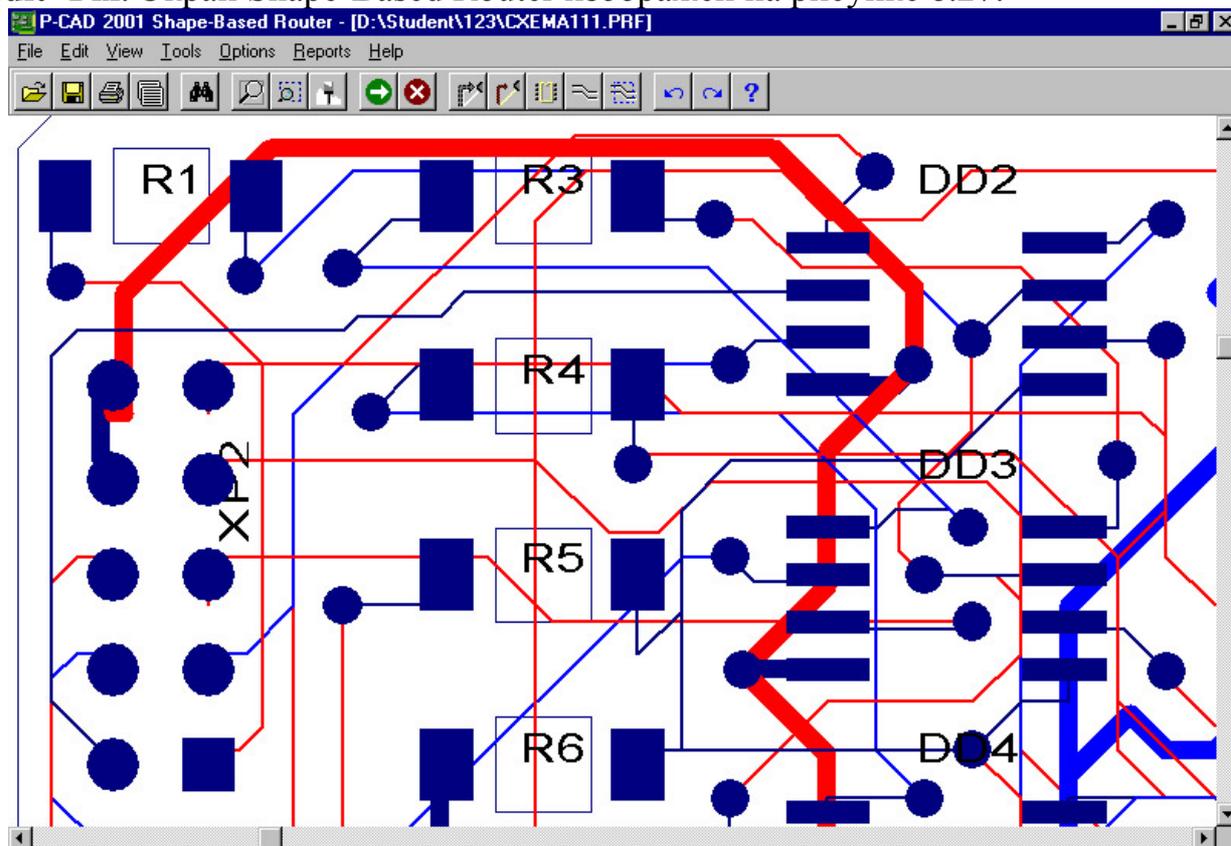


Рисунок 8.27 – Экран программы Shape-Based Router

**Настройка стратегии трассировки.** По команде **Options>Auto-Router** открывается диалоговое окно настройки стратегии трассировки (рисунок 8.28),

имеющее 3 закладки. На закладке *Routing Passes* в полях Router Passes и Manufacturing Passes выбирается тип трассировки:

**Memory** – трассировка типа «память»;

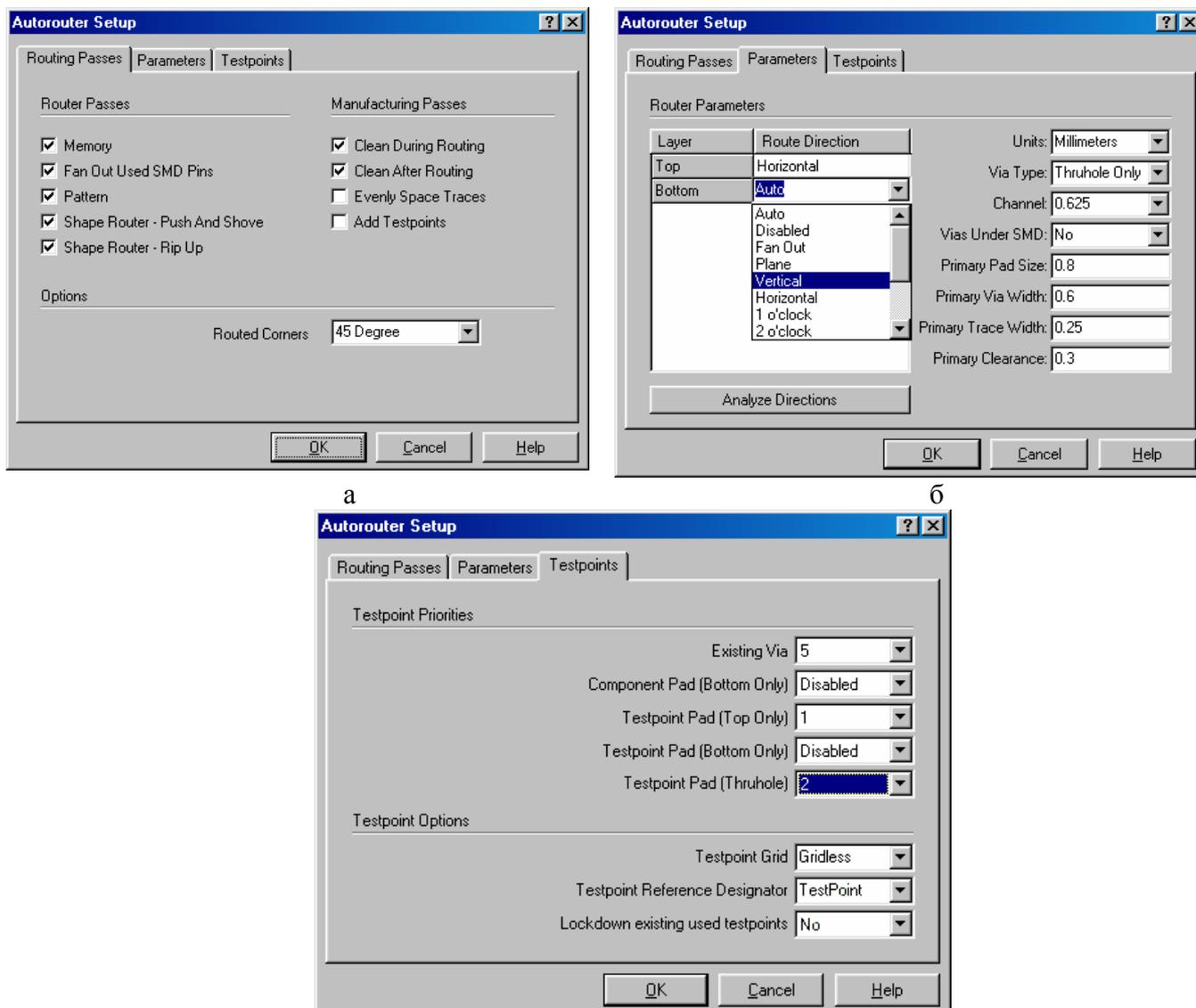


Рисунок 8.28 – Диалоговое окно Autorouter Setup: а – закладка Routing Passes, б – закладка Parameters, в – закладка Testpoints

**Fan Out User SMD Pins** – генерация веерообразно расположенных стрингеров для выводов планарных компонентов;

**Pattern** – трассировка фрагментов ПП с использованием типовых образцов, имеющих в программе (рекомендуется включать всегда);

**Shape Router – Push And Shove** – раздвигание и отталкивание ранее проложенных мешающих трасс;

**Shape Router – Rip Up** – разрыв и повторная трассировка трасс, проложенных на предыдущих проходах с нарушениями (превышение допустимых зазоров или пересечение на одном слое);

**Clean During Routing** – уменьшение количества изгибов трасс и удаление

лишних ПО в процессе трассировки;

**Clean After Routing** – уменьшение количества изгибов трасс и удаление лишних ПО после трассировки;

**Evenly Space Traces** – равномерное распределение трасс;

**Add Testpoints** – вставка контрольных точек.

В поле Options выбирается тип изгибов трасс Routed Corners: под углом 90 или 45°.

На закладке *Parameters* в поле Router Direction для каждого слоя ПП выбирают предпочтительную ориентацию трасс из следующего перечня:

**Auto** – предоставление выбора программе;

**Disable** – запрет для трассировки;

**Fan Out** – генерация веерообразно расположенных ПО, соединенных с выводами планарных компонентов короткими отрезками трасс (стрингерами);

**Plane** – резервирование слоя для использования в качестве слоя металлизации;

**Vertical** – вертикальная;

**Horizontal** – горизонтальная;

**1 o'clock, 2 o'clock, 4 o'clock, 5 o'clock, 45/, 45\|** – выбор одной из ориентации (разрешается для многослойных ПП);

**Any Direction** – отказ от прокладки трасс определенной ориентации (разрешается для многослойных ПП).

Нажатие на панель Analyze Directions выбирает предпочтительное направление трассировки слоев, имеющих признак Auto. Кроме того, назначаются следующие параметры (рис. 3.6, б):

**Units** – система единиц величин, отображаемых на экране дисплея (Mils, Inches, Centimeters, Millimeters, Microns; на точность трассировки не влияет);

**Via Type** – запрет (No Vias), разрешение (Thruhole Only) использования ПО;

**Via under SMD** – запрет (No), разрешение (Yes) размещения ПО под КП планарных компонентов;

**Channel Size** – размер канала трассировки, равный сумме ширины трассы Primary Trace Width и допустимого зазора Primary Clearance. Это значение может быть изменено; для выполнения *бессеточной* трассировки назначается 1 mil и менее.

**Primary Pad Size** – диаметр большинства КП выводов штыревых компонентов. Если штыревых компонентов нет, то этот параметр равен наибольшему размеру КП планарных компонентов (используется при расчете Channel Size);

**Primary Via Width** – диаметр ПО, выбранного в PCAD PCB в качестве ПО по умолчанию;

**Primary Trace Width** – ширина большинства трасс ПП (устанавливается программой на основе анализа таблицы цепей меню команды Edit>Net; используется при расчете Channel Size и прокладке трасс, которым в PCAD PCB не был назначен атрибут Width);

**Primary Clearance** – минимально допустимый зазор между любыми объек-

тами, т.е. Track to Track, Track to Pad, Track to Via, Via to Pad (устанавливается в PCAD PCB, но может быть изменен).

Для облегчения доработки ПП после ее трассировки рекомендуется не изменять значение Channel Size, рассчитанное программой. Бессеточная трассировка (с шагом сетки 1 mil и менее) целесообразна в проектах, имеющих большое разнообразие значений ширины трасс и зазоров. При использовании значения Channel Size, установленного программой, нет разницы между качеством сеточной и бессеточной трассировки.

На закладке *Testpoints* (рисунок 8.28, в) устанавливается приоритет размещения контрольных точек (КТ), размещаемых на ПП, если на закладке *Routing Passes* отмечена опция Add Testpoints.

По команде **Edit>Net Attributes** открывается диалоговое окно (рисунок 8.29) задания атрибутов всем цепям проекта:

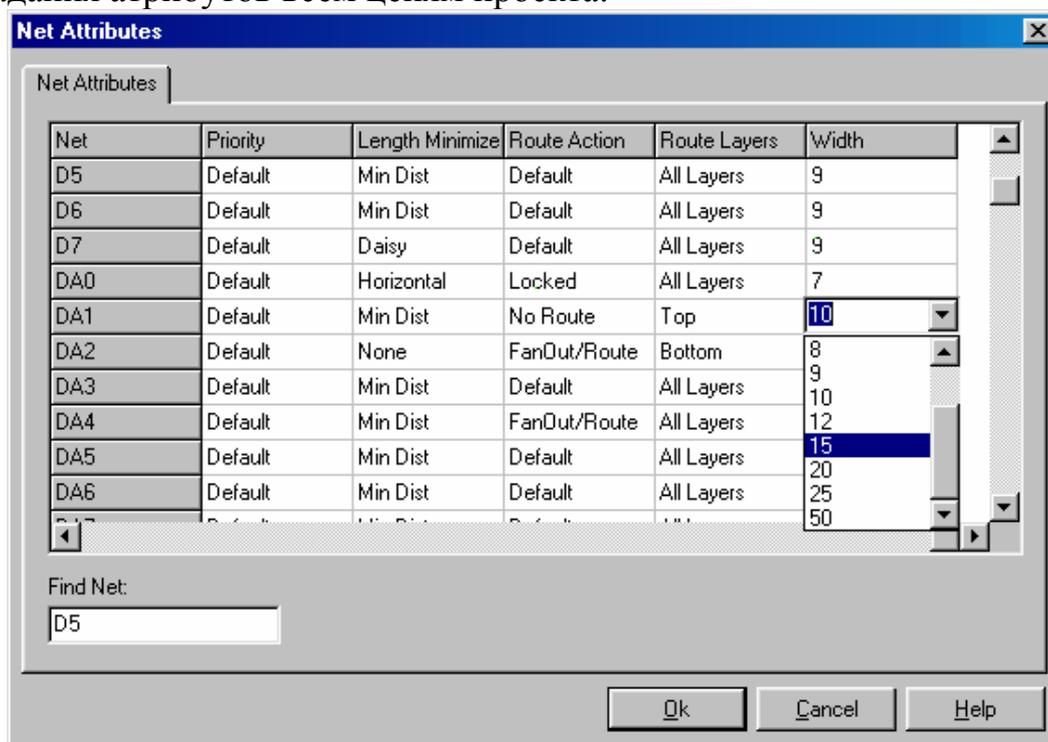


Рисунок 8.29 – Диалоговое окно Net Attributes

**Display** – разрешение (True) или запрет (False) отображения цепи;

**Routing Priority** – приоритет трассировки (Default, 1, 2, ..., 20);

**Length Minimize** (минимизация длины трассы цепи);

**Route Action** (тип трассировки);

**Routing Layer** (слой трассировки);

**Width** – ширина трассы. По умолчанию указывается значение, равное параметру Primary Trace Width; однако оно может быть изменено (выбрано из предлагаемого набора или введено самостоятельно).

Перед выполнением трассировки полезно ознакомиться со статистическими данными о ПП по команде **Reports>Pre-Route Synopsis** для внесения при необходимости коррекции в исходную ПП или стратегию трассировки. Полезная информация о плотности связей ПП может быть получена также по коман-

де **View>Density**.

Автотрассировка всей ПП начинается по команде **Tools>Start Autorouter** . Интерактивная трассировка выполняется по другим командам меню **Tools: AutoRoute Connection** , **AutoRoute Net** , **AutoRoute Component** , **AutoRoute Area** . Ручная прокладка трассы выполняется по команде **Manual Route** . С помощью команды **Tools>Sketch Route** курсором рисуется примерное расположение трассы выбранной цепи, которая прокладывается затем автоматически.

По завершении трассировки по командам **Reports>Routing Statistics**, **Reports>Reports** можно просмотреть итоговые отчеты.

Возвращение в PCAD PCB после трассировки производится по команде **File>Save and Return**.

## ЧАСТЬ 2. ВЕСЕННИЙ СЕМЕСТР

### 1 ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ В САПР

#### 1.1 Классификация задач конструкторского проектирования

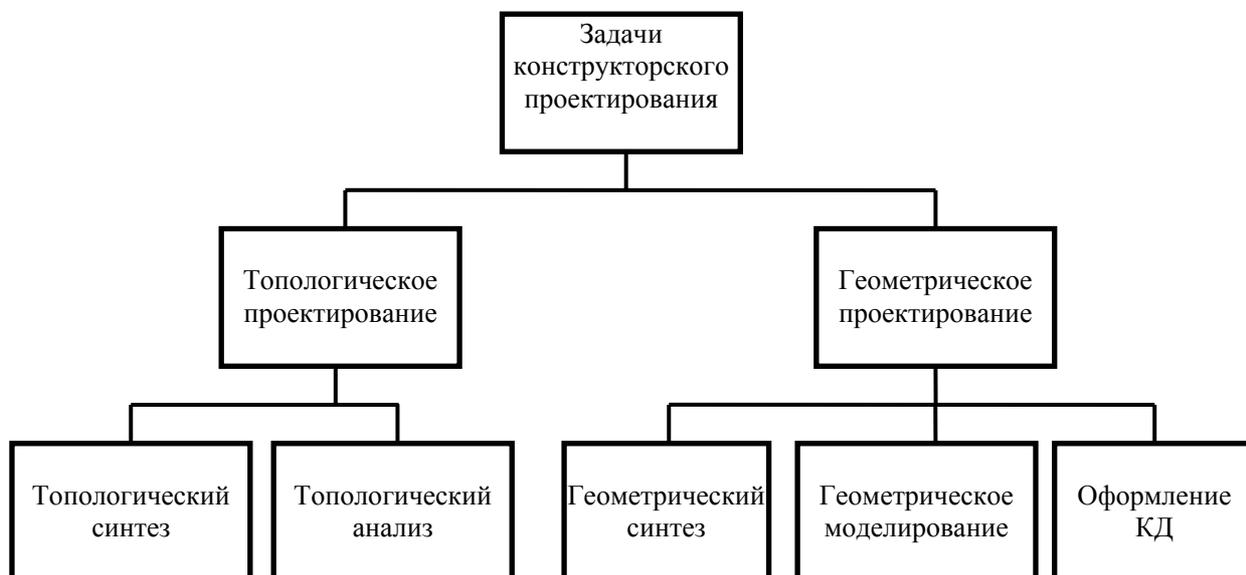


Рисунок 1.1 – Классификация задач конструкторского проектирования

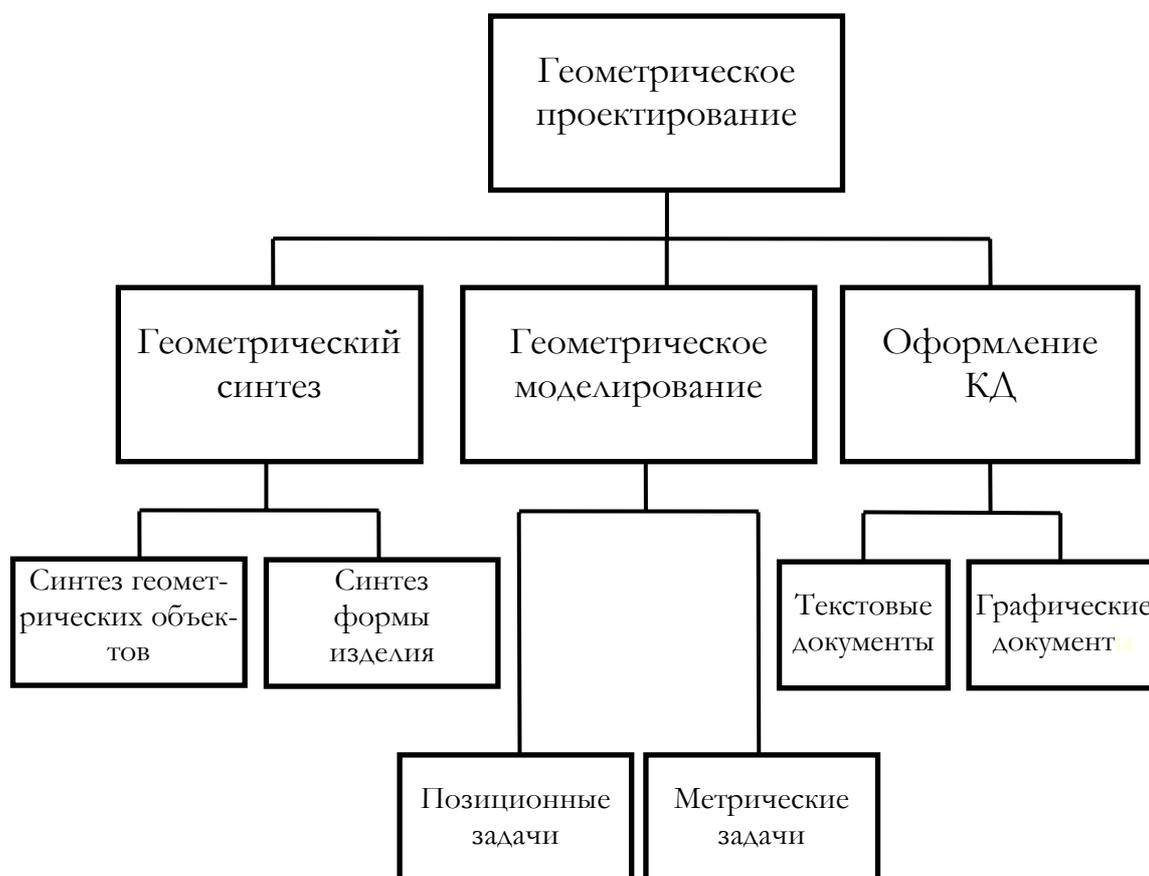


Рисунок 1.2 – Классификация задач геоматрического проектирования

## 1.2 Геометрический синтез

1) Синтез геометрических объектов:

Формирование (компоновка) сложных геометрических объектов (ГО) и элементарных ГО заданной структуры

Основной критерий – точность воспроизведения

2) Синтез формы изделия:

Получение рациональной или оптимальной формы деталей, узлов и т.п., влияющей на качество функционирования объектов конструирования

Используется на ранних стадиях проектирования изделий

## 1.3 Геометрические модели

**Геометрическая модель** – совокупность сведений, однозначно определяющих форму геометрического объекта

**Формы представления:**

- Совокупность уравнений линий и поверхностей;
- Алгебраические соотношения;
- Графы;
- Списки;
- Таблицы;
- Описания на специальных графических языках

**Применение геометрических моделей:**

- Описание геометрических свойств объекта (форма, расположение в пространстве);
- Решение геометрических задач (позиционных и метрических);
- Преобразования формы и положения геометрических объектов;
- Ввод графической информации;
- Оформление КД

**Виды геометрических моделей**

### 1. Аналитические

Представляются уравнениями, описывающими контуры или поверхности  
Служат для описания примитивов и составных объектов

### 2. Алгебраические

Геометрический объект описывается логической функцией условий, выражающих принадлежность точки какой-либо области

### 3. Канонические

Представляются уравнениями в каноническом виде

### 4. Рецепторные

Приближенное представление объекта в плоскости или пространстве рецепторов

### 5. Каркасные

Представляются каркасами геометрических фигур

### 6. Кинематические

Параметрическая форма записи для описания объектов (простые модели – поверхности вращения и т.п.)

### 7. Макромодели

Математическое описание типовых геометрических фигур

## 1.4 Комплексный подход к автоматизированному проектированию

Принципы системного подхода:

- Рассмотрение частей сложной системы с учетом их взаимодействия
- Выявление структуры системы
- Типизация связей
- Определение атрибутов
- Анализ влияния внешней среды

По сложности автоматизации проектирования.

- 1) одноэтапные;
- 2) многоэтапные;
- 3) комплексные (все этапы)

Жизненный цикл изделий производства (рисунок 1.3):

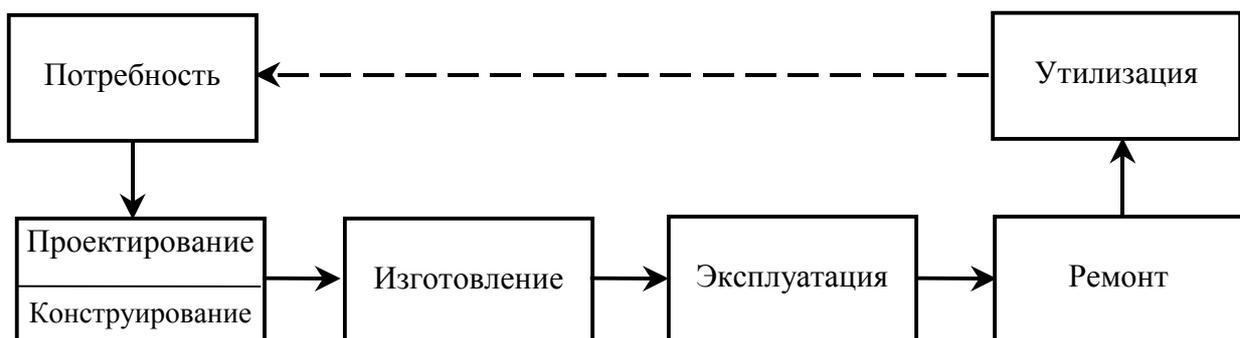


Рисунок 1.3 – Жизненный цикл изделий производства

## 1.5 Автоматизированные CAD/CAM/CAE/PDM комплексы

### **Классификация пакетов САПР:**

- 1) САПР нижнего уровня (CAD) (AutoCAD, Компас ..)
- 2) САПР среднего уровня (CAD/CAM)(T-FLEX CAD, SolidWorks, Техно-Про, Autodesk Mechanical Desktop ..)
- 3) САПР высокого уровня (CAD/CAM/CAE/PDM) (Pro/Engineer, T-FLEX..)

### **1. Программы CAD:**

- программное обеспечение для создания чертежей и трехмерных моделей, а также программы для инженеров-технологов (составление технологических процессов)

### **2. Программы CAM:**

- программное обеспечение для автоматического и полуавтоматического создания и редактирования управляющих программ для станков с ЧПУ, а также ПО для передачи управляющих программ на станки с ЧПУ.

### **3. Программы CAE:**

- программное обеспечение для инженерных расчетов. ПО для общих и специализированных расчетов.

### **4. Программы PDM:**

- программное обеспечение для ведения документооборота, создания и управления архивами чертежей, а также ПО для работы со сканированными документами технического назначения..

**CIM (COMPUTER INTEGRATED MANUFACTURING)** – системы, обеспечивающие автоматизацию всего жизненного цикла изделия, начиная с задач маркетинга, продолжая проектированием и производством, завершая сбытом продукции, сервисным обслуживанием и утилизацией изделий

### **CALS-технологии – путь к комплексной автоматизации**

- Объединение в едином комплексе задач 1-4
- «поддержка жизненного цикла» изделия
- «повышение конкурентоспособности за счет сокращения затрат, сокращения сроков вывода новых образцов на рынок, повышения качества продукции за счет сквозной поддержки ее жизненного цикла»

### **Например**

T-FLEX CAD 2D, T-FLEX CAD 3D, T-FLEX DOCs, T-FLEX Технология, T-FLEX ЧПУ 2D, T-FLEX ЧПУ 3D, T-FLEX Анализ – одно целое.

## **2 СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ**

### **AutoCAD 2000/2000i/2002/2004**

#### **2.1 Назначение и возможности САПР AutoCAD**

AutoCAD (Automated Computer Aided Design - автоматизированное компьютерное проектирование) – универсальная система автоматизированного проектирования (САПР) фирмы Autodesk (США), предназначенная для разработки двухмерных чертежей и рисунков, а также разработки и моделирования каркасных, полигональных (поверхностных) и объемных (твердотельных) конструкций в различных областях человеческой деятельности (техника, строительство и архитектура, швейное производство и т.п.).

Чертежи (рисунки) системы AutoCAD хранятся в файлах с расширением: dwg.

#### **Основные функции графического редактора**

1. Создание новых чертежей и сохранение их на диск.
2. Редактирование существующих чертежей.
3. Вывод чертежей на плоттер, принтер.
4. Преобразование чертежей, созданных предыдущими версиями редактора.
5. Восстановление испорченных чертежей.
6. Формирование электронного архива чертежей и "библиотек стандартных элементов"
7. Контроль стандартов предприятий с помощью специальных средства, позволяющих управлять слоями, стилями и т.п. (Начиная с AutoCAD 2002)
8. Возможность расширения и специализации системы:
  - встроенный компилятор языка AutoLISP;
  - средства разработки приложений на языке программирования СИ
9. Использоваться в локальном или сетевом варианте (Начиная с AutoCAD 2002)
10. Работать с двумя мониторами: один — для выдачи запросов на ввод команд и вывод текстов, а второй — для выполнения линий построения и редактирования чертежей

#### **Основные особенности САПР AutoCAD 2004**

- Выполнена внешняя и внутренняя интеграция системы AutoCAD с операционной системой Windows XP (Professional Edition и Home Edition).
- Введен новый, более компактный DWG-формат, который позволяет просматривать растровые образцы DWG-файлов в проводнике Windows, а также защищать паролем и цифровой подписью файлы рисунков.
- Появился новый элемент интерфейса — Tool Palettes (Палитры инструментов), который дает возможность быстрого доступа к часто используемым

элементам пользователя (блокам, штриховкам и заливкам).

- Появилась немодальная форма окон Properties (Свойства), Design Center (Центр управления), которые сворачиваются и легко раскрываются.

- Добавлена возможность расширения графического экрана.

- Введены новые панели Styles (Стили) и Draw Order (Порядок следования).

- Список Layers (Слои) и кнопки операций над слоями выделены из панели инструментов Properties (Свойства) и стали самостоятельной панелью.

- Количество используемых цветов увеличилось с 256 до 16 миллионов.

- Появилась возможность печати раскрашенных и тонированных видовых экранов.

- Введена возможность настройки строки режимов (убирать ненужные кнопки и восстанавливать нужные).

- Значительно улучшены средства создания и редактирования многострочных текстов (в частности, введены позиции табуляции).

- Появились градиентные заливки.

- Улучшился механизм работы с файлами внешних ссылок.

- Появилась возможность более тонкой настройки функций правой кнопки мыши, в зависимости от продолжительности щелчка.

- Для инсталляции AutoCAD 2004 не требуется удалять с компьютера предыдущую версию AutoCAD (2002 или более раннюю).

- Система AutoCAD 2004 не может быть установлена в среде операционной системы Windows 95.

- AutoCAD 2004 работает в операционной системе Windows 98 с предварительно установленным Internet Explorer 6.0 и, видимо, в Windows Me, хотя официально не рассчитан на них. Дилеры Autodesk не принимают по ним вопросов. Начиная с AutoCAD 2004 фирма Autodesk полностью прекращает поддерживать в своих продуктах платформу Windows 98, 98SE, Me.

- AutoCAD 2004 работает в операционных системах: Windows 2000; Windows XP (Professional Edition); Windows XP (Home Edition); Windows NT 4.0 (с обновлением Service Pack 6a или более поздним).

### **Требования к компьютеру**

Требования к компьютеру различны в зависимости от версии программного обеспечения. Для каждой более поздней версии AutoCAD требования к компьютеру ужесточаются.

Система AutoCAD 2004 может использоваться в локальном или сетевом варианте. Для ее нормального функционирования требуется получить специальный код авторизации от фирмы Autodesk. Первые версии AutoCAD содержали в основном инструменты для простого двумерного рисования, которые постепенно дополнялись и развивались. В результате система стала очень удобным "электронным кульманом". На рисунке 2.1 приведен пример двумерного чертежа, сделанного сравнительно простыми средствами AutoCAD. Для формирования такого чертежа не требуется работа с трехмерной моделью изделия.

Таблица 2.1 – Требования к компьютеру

AutoCAD 2000	AutoCAD 2004
<ul style="list-style-type: none"> <li>- процессор не хуже P 133,</li> <li>- рекомендуемый объем памяти – 64 Мб (минимальный – 32 Мб),</li> <li>- жесткий диск – не менее 130 Мб свободного пространства, (50 Мб свободного дискового пространства в системном каталоге, не менее 64 Мб в файле подкачки)</li> </ul> <p>Система проектирования занимает порядка 150–190 Мб (в зависимости от варианта установки).</p> <p>Необходимо наличие мыши, монитор SVGA с разрешением не хуже 800x600</p>	<ul style="list-style-type: none"> <li>- компьютер не ниже Pentium III с процессором 500 МГц,</li> <li>- рекомендуемый объем памяти – 256 Мб,</li> <li>- жесткий диск – 2 Гб;</li> </ul> <p>на винчестере надо иметь свободными 350Мб под программное обеспечение и не менее 250 Мб для временных файлов</p> <p>Необходимо наличие мыши, монитор SVGA с разрешением не хуже 800x600</p>

Большим преимуществом системы AutoCAD, как средства рисования, является возможность последующего формирования электронного архива чертежей. Каждый из созданных таким образом файлов рисунков легко редактируется, что позволяет быстро получать чертежи-аналоги по чертежам-прототипам. В качестве средств защиты от несанкционированного доступа в файлах рисунков, созданных системой AutoCAD 2004, предусмотрены пароли и электронные цифровые подписи.

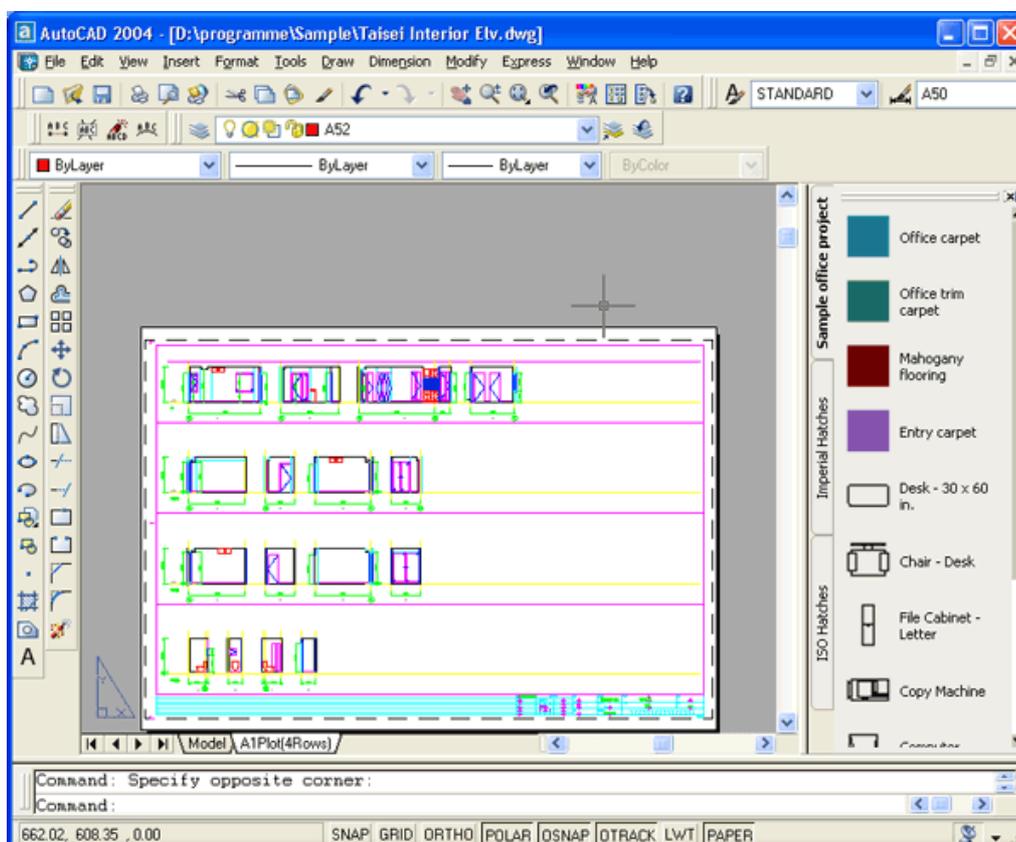


Рисунок 2.1 – Двумерный чертеж

Для облегчения процесса выпуска проектной документации можно разрабатывать "библиотеки стандартных элементов". В качестве стандартных элементов могут выступать как целые файлы, так и их отдельные части. Эта идея стала хорошим стимулом для создания на базе системы AutoCAD локальных рабочих мест по различным конструкторским, архитектурным и другим направлениям, а также для разработки новых специализированных систем. Мощным дополнением к этому является возможность использования языков программирования.

Начиная с AutoCAD 2002, в систему включены специальные средства для контролирования стандартов предприятий, позволяющих управлять слоями, стилями и т. п.

Уже десятая версия AutoCAD позволяла выполнить достаточно сложные трехмерные построения в любой плоскости пространства и отобразить их на разных видовых экранах с различных точек зрения. Поэтому она стала также инструментом и трехмерного моделирования. Механизм пространства листа и видовых экранов дал возможность разрабатывать чертежи с проекциями трехмерных объектов или сооружений. В системе AutoCAD по одной модели можно получить несколько листов чертежного документа. На рисунке 2.2 приведен пример чертежа, сформированного как отображение проекций и сечений трехмерного объекта.

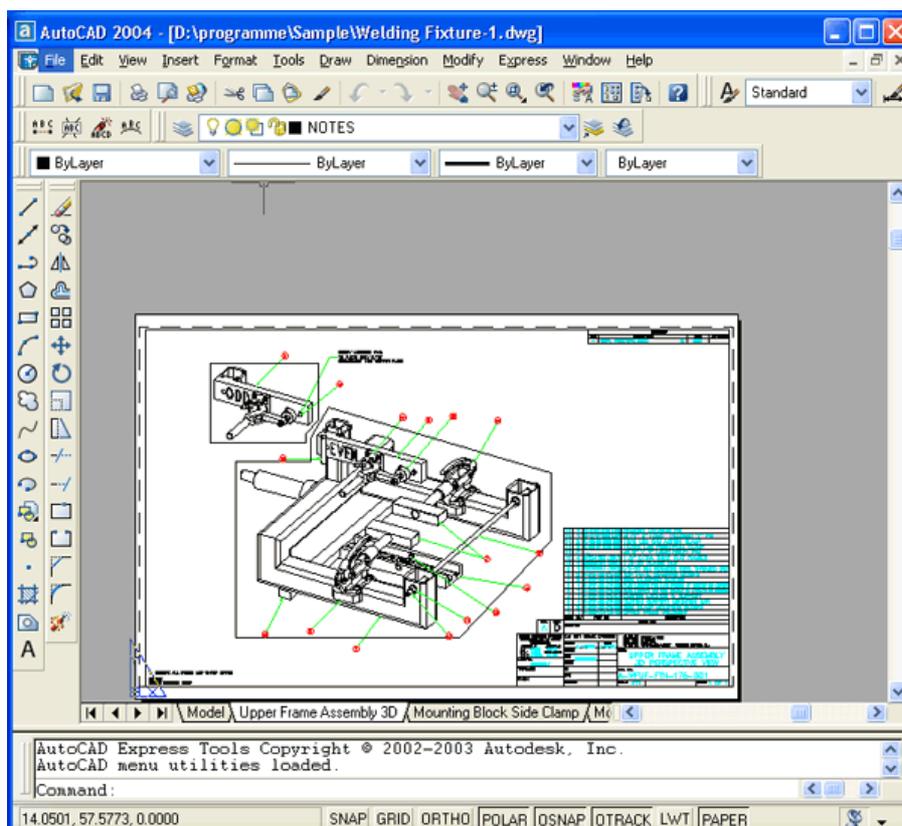


Рисунок 2.2 – Чертеж части устройства дверного замка

На рисунке 2.3 показана модель корпуса судна, являющаяся основой для последующего формирования чертежа и построения контуров деталей. Степень

проработанности, а также сложность и насыщенность такой модели могут быть разными и определяются тем, как она будет в дальнейшем использована.

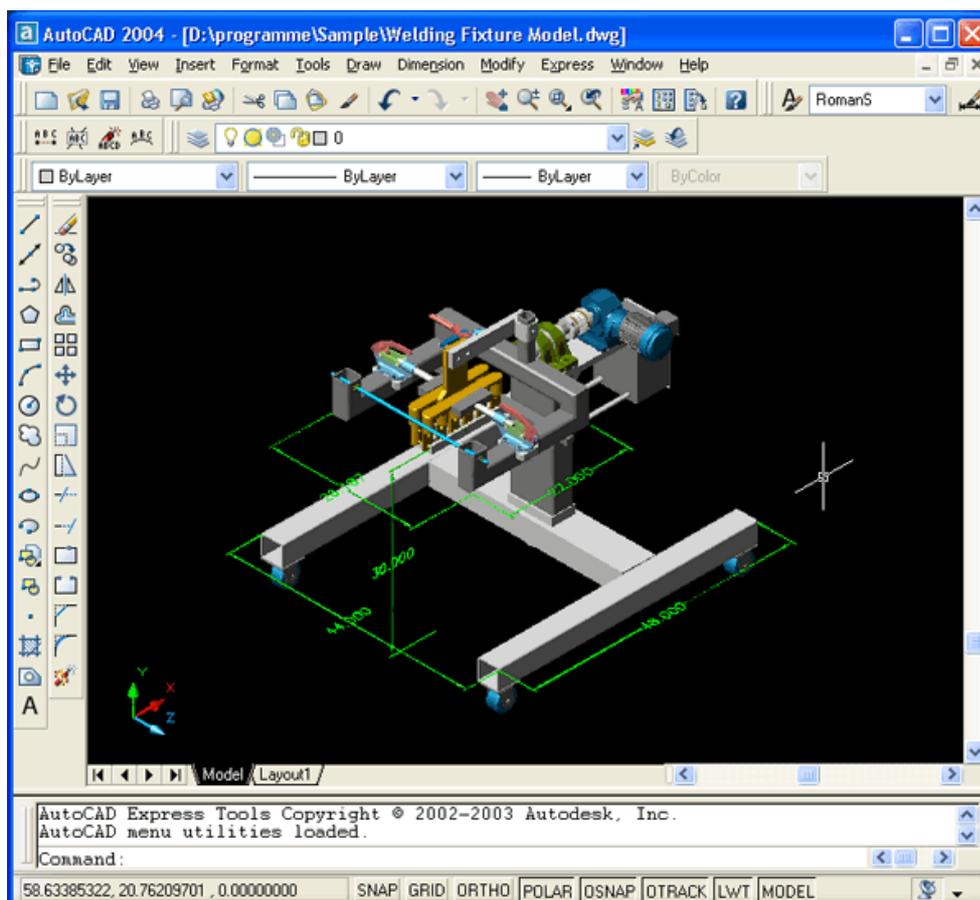


Рисунок 2.3 – Трехмерная модель платформы

## 2.2 Пользовательский интерфейс

Итак, вы установили систему AutoCAD 2004 и готовы начать работу. Запуск системы осуществляется с помощью двойного щелчка левой кнопки мыши по ярлыку , расположенному на рабочем столе. Другой способ запуска — использование меню Пуск (Start) рабочего стола Windows (Пуск | Программы | Autodesk | AutoCAD 2004 | AutoCAD 2004).

Сразу после старта системы открывается стандартное диалоговое окно Startup (Начало работы), в котором необходимо выбрать режим работы. В системах AutoCAD 2000i и AutoCAD 2002 был другой вариант начала — диалоговое окно AutoCAD Today (Проектный центр), который был доступен помимо стандартного. В рассматриваемую версию системы AutoCAD этот вариант не включен.

В диалоговом окне Startup (Начало работы) пользователю необходимо подвести указатель мыши к одной из больших кнопок с режимами начала работы, изображенных в верхней части окна:

-  Open a Drawing (Открытие рисунка);
-  Start from Scratch (Простейший шаблон);
-  Use a Template (По шаблону);
-  Use a Wizard (Вызов мастера).

Затем следует щелкнуть левой кнопкой мыши. После этого выбранная вами большая кнопка изменит свое состояние на "нажатая", а содержимое текста надписи, расположенной в правой верхней части окна, модифицируется на название кнопки. Для завершения работы с диалоговым окном остается только щелкнуть левой кнопкой мыши по кнопке ОК.

Если вам необходимо создать новый пустой рисунок, то в окне Startup (Начало работы) нужно воспользоваться кнопкой Start from Scratch (Простейший шаблон). Этот режим предлагается по умолчанию. В нем создается новый пустой чертеж с минимумом установок. Пользователю только нужно отметить, в каких единицах измерения он будет работать: Imperial (feet and inches) (Английские (футы и дюймы)) или Metric (Метрические). Второй вариант, конечно, в этом случае для нас предпочтительнее.

При выборе режима Use a Template (По шаблону) пользователь имеет возможность начать работу не с нуля, а с каким-то наполнением нового рисунка (это могут быть рамки, штампы и какие-то другие линии или установки).

Режим Use a Wizard (Вызов мастера) дает возможность вызвать одну из программ-мастеров настройки нового рисунка. Будем считать, что студент для первого раза воспользуется режимом по умолчанию Start from Scratch. В открывшемся окне (рисунок 2.4) разберем элементы оформления рабочего окна AutoCAD.

Центральная часть экрана — это основная рабочая зона, в которой находится видимая часть рисунка (остальные его части могут находиться выше, правее, ниже и левее). При движении курсора по этой части, которую мы будем называть далее графическим экраном, он (курсор) имеет вид перекрестия с квадратной мишенью в точке пересечения (на рис. 2.4 — обозначен номером 12). Длина линий перекрестия настраивается с помощью системной переменной CURSORSIZE.

В левом нижнем углу графического экрана находится знак осей системы координат (обозначенный цифрой 4). При необходимости знак может быть настроен или приведен к виду, который он имел в AutoCAD 2000.

Верхнюю строку экрана (1), состоящую из надписей: File (Файл), Edit (Правка), View (Вид), Insert (Вставка), Format (Формат), Tools (Сервис), Draw (Рисование), Dimension (Размеры), Modify (Редакт), Express (Экспресс), Window (Окно), Help (Справка) — будем называть строкой меню. Эта строка содержит наименования падающих меню, раскрыть любое из которых можно, выбрав с помощью мыши соответствующее имя меню.

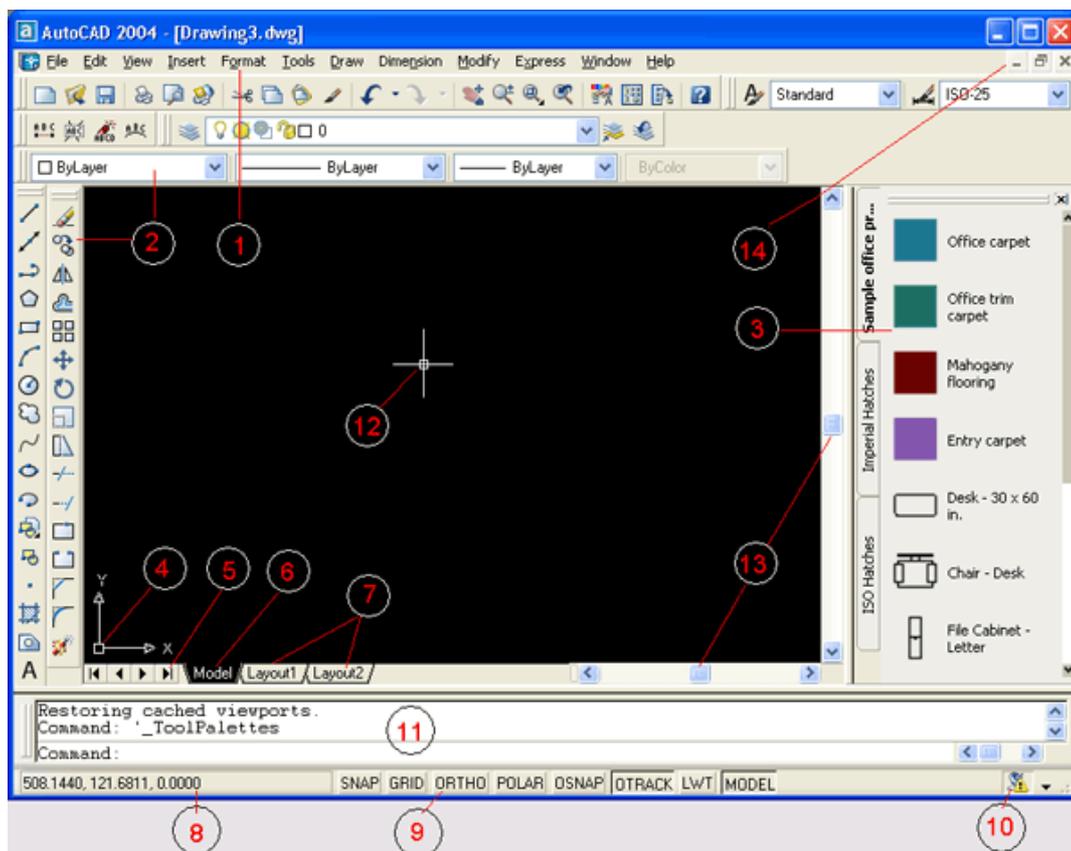


Рисунок 2.4 – Элементы пользовательского интерфейса: 1 — строка меню, 2 — панели инструментов, 3 — панель Tool Palettes, 4 — пиктограмма осей координат, 5 — кнопки показа списка вкладок, 6 — вкладка пространства модели, 7 — вкладки листов, 8 — строка состояния со счетчиком координат, 9 — кнопки режимов, 10 — Communication Center, 11 — зона командных строк, 12 — курсор, 13 — линейки прокрутки, 14 — кнопки управления окном документа

Графический экран снизу обрамляют кнопки вкладок Model (Модель) (элемент 6), Layout 1 (Лист!) и Layout 2 (Лист2) (элементы 7). Вкладки используются при переключении между пространствами модели и листа (см. гл. 10). По умолчанию активной является вкладка Model (Модель). Треугольные кнопки (5) слева от имен вкладок позволяют передвигаться по списку вкладок в обоих направлениях (аналогично движению по списку листов в книге Microsoft Excel).

Справа от вкладок расположена горизонтальная линейка прокрутки для графического экрана. Вертикальная линейка прокрутки расположена по правой границе графического экрана. Обе линейки прокрутки на рис. 2.4 обозначены номером 13.

Нижняя часть экрана (11), в которой вы на рис. 2.4 видите приглашение в форме Command: (Команда:) — это область, через которую в основном происходит диалог пользователя с системой, здесь отображаются вводимые вами команды и ответы (или вопросы) AutoCAD. Эту часть будем называть зоной командных строк. Последняя строка, содержащая приглашение Command: (Команда:), называется командной строкой.

Ниже от зоны командных строк находится строка состояния (8), в которой

расположены счетчик координат и прямоугольные кнопки режимов (9): SNAP (ШАГ), GRID (СЕТКА), ORTHO (ОПТО), POLAR (ОТС-ПОЛЯР), OSNAP (ПРИВЯЗКА), OTRACK (ОТС-ПРИВ), LWT (ВЕС), MODEL (МОДЕЛЬ). Счетчик координат служит для ориентировки на поле чертежа — он изменяет свое трехкоординатное значение при движении указателя мыши по графическому экрану. Счетчик может быть отключен с помощью функциональной клавиши <F6>. Нажатием на ту же клавишу счетчик можно снова включить. Другой способ включения и выключения счетчика — щелчок левой кнопкой мыши по строке счетчика.

Кнопка  (10), появившаяся только в версии AutoCAD 2004, предназначена для вызова диалогового окна Communication Center (Центр связи), которое дает возможность при наличии соединения с Интернетом оперативно получать сведения об обновлениях, выпущенных фирмой Autodesk. Работа с этим окном рассматривается в приложении 3.

Основным элементом пользовательского интерфейса являются кнопки панелей инструментов. Четыре горизонтальные панели на рис. 2.4 находятся ниже строки меню. Еще две вертикальные панели находятся слева от графического экрана.

Все панели инструментов в системе AutoCAD 2004 оформлены в стиле Windows XP (с использованием серо-голубой гаммы цветов). Если вы подведете курсор к какой-нибудь пиктограмме любой панели, то пиктограмма сразу примет форму прямоугольной кнопки, а через несколько мгновений под указателем появится всплывающая подсказка с наименованием команды или функции AutoCAD, выполняемой с помощью этой кнопки. На рис. 2.5 показана подсказка, появляющаяся при положении верхней точки указателя мыши над кнопкой команды LINE (ОТРЕЗОК).

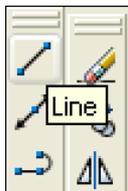


Рисунок 2.5 – Всплывающая подсказка

Одновременно в строку состояния выводится справочная информация к команде Creates straight line segments: LINE (Создает прямолинейные отрезки: ОТРЕЗОК).

Шесть панелей, которые показаны на рис. 2.4, находятся в фиксированном положении. При желании можно перенести их внутрь графического экрана, где они займут "плавающее" положение, принимая одновременно форму окон Windows (рисунок 2.6). Для этого нужно аккуратно подвести указатель мыши к краю панели (но не задевая кнопок инструментов!), нажать левую кнопку мыши и, не отпуская ее, перетащить панель на свободное место экрана, после чего отпустить левую кнопку.

В плавающем положении панели могут принимать разную форму, которую

можно изменять, растягивая их за углы. Для перемещения панели на новое место следует перетащить ее с помощью мыши за синюю область с наименованием. Перемещение панели на один из четырех краев графического экрана позволяет вернуть ее к фиксированному положению. Таким образом, пользователь может сам найти для себя наиболее удобный способ расположения и формы панелей.

На рисунках 2.4, 2.6 под строкой меню видны четыре горизонтальные панели с наименованиями: Standard (Стандартная) (в первом ряду сверху), Layers (Слои) и Styles (Стили) (во втором ряду сверху) и Properties (Свойства) (в третьем ряду сверху). Эти панели желательно не трогать, не удалять и не перемещать, поскольку они постоянно нужны для работы. По сравнению с предыдущими версиями в системе AutoCAD 2004 появилась новая панель Styles (Стили), а панель Object Properties (Свойства объектов) разделилась на панели Layers (Слои) и Properties (Свойства).

Две панели, вертикально расположенные на рисунке 2.4, называются Draw (Рисование) и Modify (Редактирование).

Для вызова панелей на экран и удаления их с экрана имеются специальные средства. Подведите курсор к имени падающего меню View (Вид) и нажмите левую кнопку мыши. Выбранное падающее меню откроется (выпадет) внутрь графического экрана (рисунок 2.7). Затем опустите указатель мыши на нижнюю строку Toolbars (Панели) и нажмите левую кнопку мыши (в дальнейшем выбор любых элементов экрана будем делать только с помощью щелчка левой кнопки мыши, правая же кнопка обычно вызывает какое-нибудь контекстное меню, о которых речь пойдет дальше). Многоточие после наименования пункта Toolbars (Панели) означает, что после выбора этого пункта меню будет вызвано диалоговое окно, в котором следует ввести какие-то параметры и ответить на предложенные вопросы.

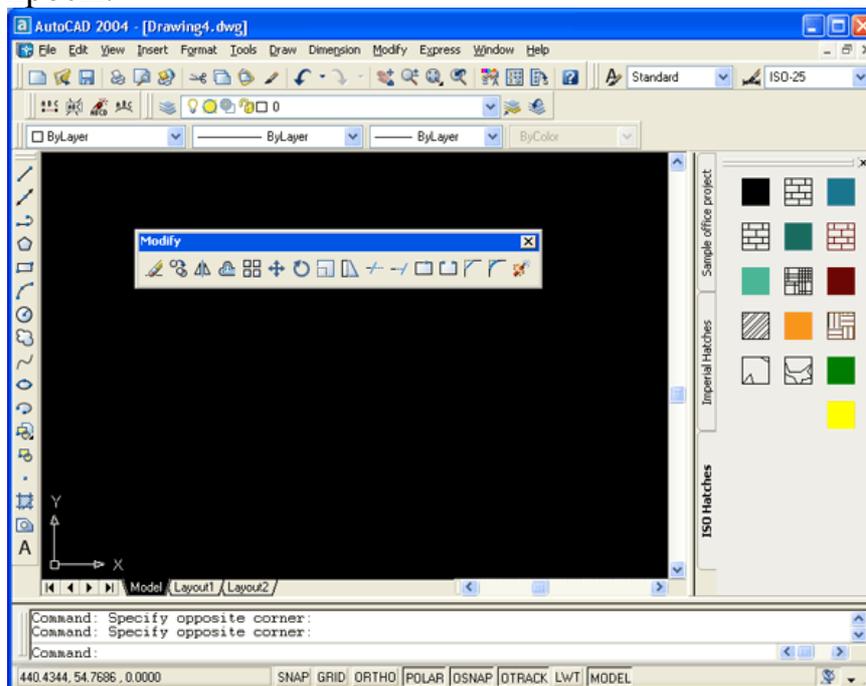


Рисунок 2.6 – Плавающее положение панели инструментов Modify

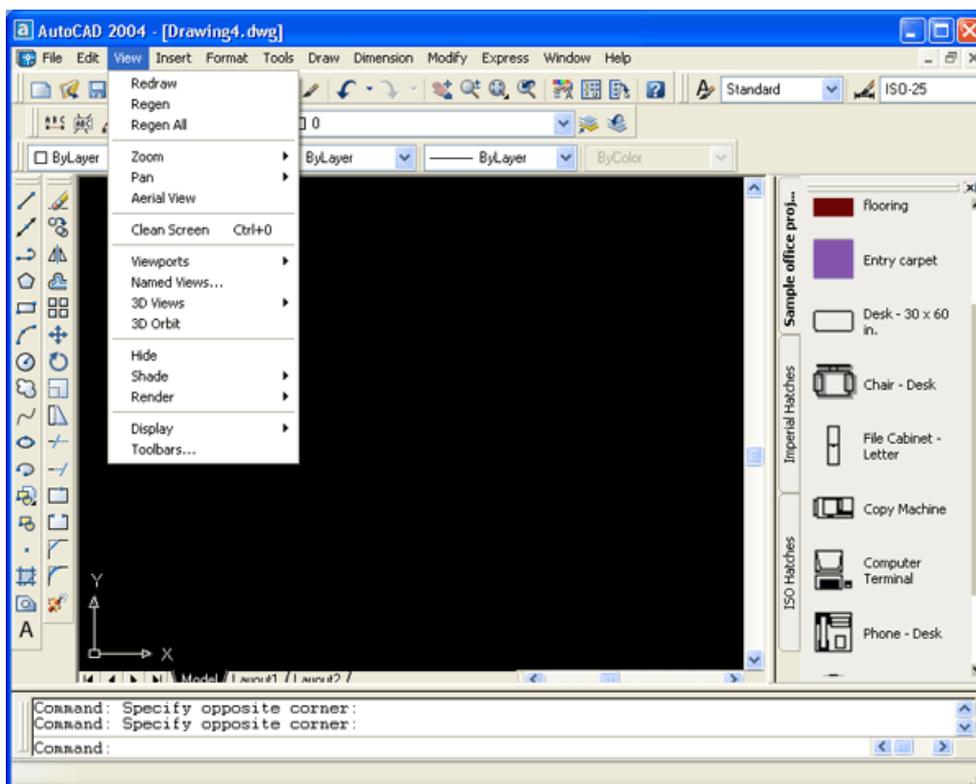


Рисунок 2.7 – Вызов падающего меню View

Появившееся диалоговое окно Customize (Адаптация) (рисунок 2.8) при выборе вкладки Toolbars (Панели) показывает в левом вертикальном списке все имена панелей, которые доступны в данной версии системы AutoCAD. Прокручивание всего списка осуществляется с помощью вертикальной линейки прокрутки. При установленном флажке в квадрате перед именем панели, панель будет находиться на экране, при отсутствии этого флажка панели на экране не будет.

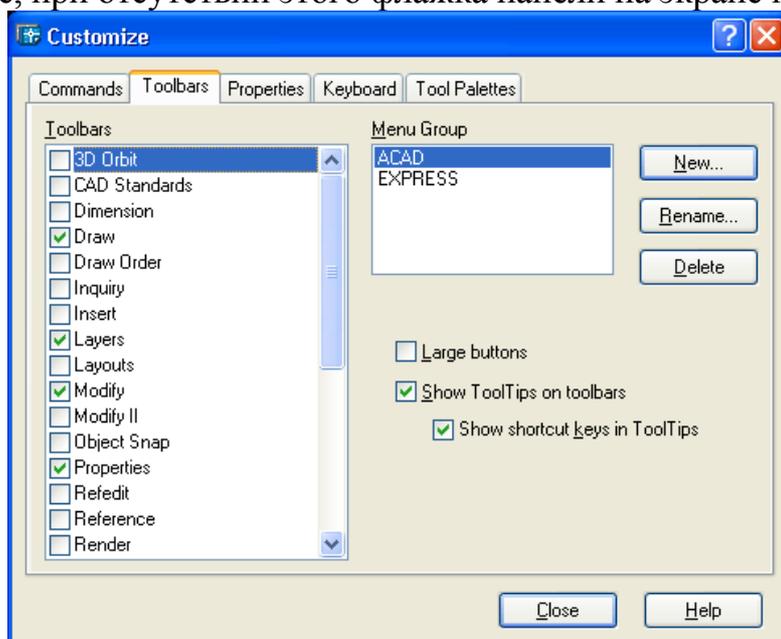


Рисунок 2.8 – Диалоговое окно Customize (вкладка Toolbars)

Установите с помощью левой кнопки мыши флажок возле панели

Dimension (Размеры) и снимите его около панели Draw (Рисование). После этого вертикально расположенная панель Draw (Рисование) исчезнет, а панель Dimension (Размеры) появится на экране (рисунок 2.9).

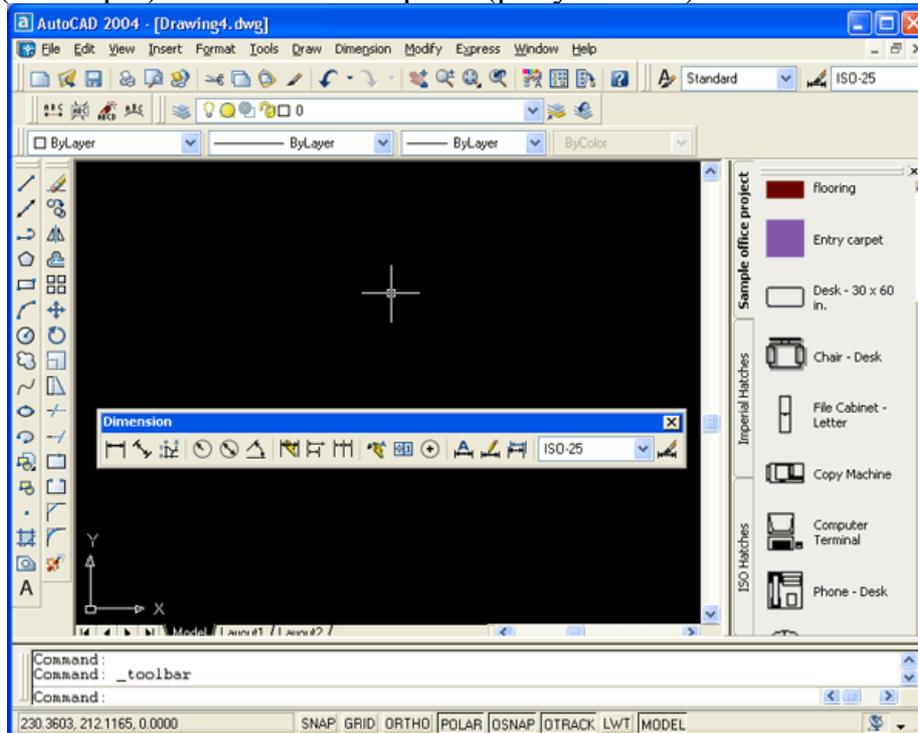


Рисунок 2.9 – Добавленная панель Dimension

Закройте диалоговое окно Customize (Адаптация) с помощью кнопки Close (Закрывать). Далее новую панель можно известным нам способом перетащить на удобное место экрана. Аналогично добавляются любые другие панели инструментов.

Если нужно вызвать на экран только одну панель или только одну панель убрать с экрана, то следует воспользоваться контекстным меню. Контекстные меню вызываются по щелчку правой кнопки мыши. Содержание контекстного меню зависит от двух условий: работает ли в данный момент какая-нибудь команда системы AutoCAD и где в данный момент находится курсор мыши. Подведите указатель мыши к какой-нибудь из находящихся на экране панелей инструментов, например, к расположенной горизонтально панели Standard (Стандартная), и щелкните правой кнопкой мыши. В том месте, где находится курсор, у вас должно появиться контекстное меню работы с панелями.

Появившееся контекстное меню содержит перечень всех панелей, а также пункт Customize (Адаптация) для вызова диалогового окна, приведенного на рис. 2.8. Если слева от имени панели стоит символ "галочка", то эта панель уже находится на экране. Щелчок левой кнопки мыши по наименованию панели изменяет ее состояние на противоположное (т. е. убирает, если она была на экране, или выводит на экран, если ее там не было).

Удалить панель с экрана можно также щелчком по кнопке  (эта кнопка видна, когда панель находится в плавающем положении). Новым элементом пользовательского интерфейса в системе AutoCAD 2004 яв-

ляется панель Tool Palettes (Палитры инструментов), которая на рисунке 2.4 обозначена цифрой 3. Вызов этой панели на экран и убирание ее с экрана осуществляется щелчком левой кнопки мыши по кнопке  панели инструментов Standard (Стандартная), а также с помощью комбинации клавиш <Ctrl>+<3> (т. е. при нажатой клавише <Ctrl> нажать клавишу с цифрой 3 и затем обе клавиши отпустить). Панель Tool Palettes (Палитры инструментов) предназначена для быстрого вызова таких наиболее употребительных элементов оформления чертежа, как штриховки и блоки.

На рис. 2.4 номером 14 обозначены кнопки управления окнами документов. Для системы AutoCAD каждый рисунок является графическим документом. Каждый документ имеет собственное окно, которое размещается внутри графического экрана. Упомянутые кнопки позволяют выполнять с активным документом такие операции, как сворачивание, восстановление и закрытие.

Еще одним новым интересным инструментом управления пользовательским интерфейсом является пункт Clean Screen (Очистка экрана) падающего меню View (Вид). С помощью этого пункта можно увеличить рабочую зону графического экрана за счет временного удаления панелей, заголовка окна AutoCAD и лотка рабочего стола Windows. В таком состоянии пользователю остаются доступными падающие меню.

Повторный выбор пункта Clean Screen (Очистка экрана) возвращает окно системы AutoCAD к обычному виду (с панелями). Для быстрого переключения между этими состояниями графического экрана можно пользоваться комбинацией клавиш <Ctrl>+<0>.

Операциям расширения и восстановления экрана соответствуют новые команды CLEANSCREENON и CLEANSCREENOFF (в момент написания данной книги русские аналоги этих команд были не известны).

## 2.3 Способы ввода команд

Система AutoCAD создана для интерактивной работы пользователя. Весь диалог с системой идет на языке команд. Самый простой способ ввода команд — это набор их на клавиатуре в командной строке в ответ на приглашение Command: (Команда:). Наименования команд в англоязычной версии вы можете вводить в латинском регистре (верхнем или нижнем). Если же вы работаете в русскоязычной версии, то можно пользоваться русскими именами команд или английскими, но предваренными знаком подчеркивания (так, например, в русскоязычной версии команду ОТРЕЗОК можно заменить командой \_LINE).

Команда LINE (ОТРЕЗОК) для рисования отрезков может быть введена в английской версии AutoCAD с клавиатуры любым из следующих способов.

- LINE
- Line
- line
- liNE

Кроме того, в английской версии обычно допускается и форма команды с символом подчеркивания (LINE, line). После набора команды на клавиатуре не забудьте нажать клавишу <Enter>, поскольку она является для системы AutoCAD указанием начать обработку команды. Пока клавиша <Enter> не нажата, вы можете отредактировать набранный в командной строке текст, используя клавиши <->, <<->, <Del> (удаляет символ справа от курсора), <Backspace> (удаляет символ слева от курсора).

Другим способом ввода команд является выбор соответствующих пунктов меню или кнопок панелей инструментов.

Если в ответ на запрос Command: (Команда:) нажать клавишу <Enter>, то AutoCAD повторит вызов предыдущей команды.

Прервать выполнение любой команды, уже начавшую свою работу, можно, нажав клавишу <Esc>.

## 2.4 Операции с файлами рисунков

Чертежи (рисунки) системы AutoCAD хранятся в файлах с расширением dwg (формат, в котором записывается графическая информация в этих файлах, называется DWG-форматом). Имена файлов могут содержать русские и латинские буквы, цифры, специальные знаки (@, #, \$, &, \_, -), а также пробелы. Остальные символы (точки, запятые и т. п.), как правило, не допускаются, поскольку являются служебными и могут быть неправильно интерпретированы операционной системой Windows. Для удобства работы желательно пользовательские чертежи хранить в отдельных папках и ни в коем случае не записывать их в основные и вспомогательные папки AutoCAD, иначе такие файлы будут не только засорять программное обеспечение, но и при смене версии или при переустановке системы AutoCAD могут быть потеряны.

Операции над файлами и некоторые другие собраны в падающем меню File (Файл)

- New (Новый)
- Open (Открыть)
- Close (Закреть)
- Partial Load (Частичная загрузка)
- Save (Сохранить)
- Save As (Сохранить как)
- eTransmit (Сформировать комплект)
- Publish to Web (Публикация в Интернете)
- Export (Экспорт)
- Page Setup (Параметры листа)
- Plotter Manager (Диспетчер плоттеров)
- Plot Style Manager (Диспетчер стилей печати)
- Plot Preview (Предварительный просмотр)
- Plot (Печать)

- Publish (Публикация)
- Drawing Utilities (Утилиты)
- Send (Переслать)
- Drawing Properties (Свойства рисунка)
- Exit (Выход)

Пункт New (Новый) падающего меню выполняет команду AutoCAD NEW (НОВЫЙ), которая создает в рабочей области новый рисунок по простейшему шаблону или по специальному шаблону — согласно выбору пользователя. При этом вызывается диалоговое окно Create New Drawing (Создание нового рисунка) (рисунок 2.10), которое похоже на окно Startup (Начало работы), но в нем недоступна кнопка открытия существующего файла. Следует заметить, что можно настроить AutoCAD таким образом, что это окно вызываться не будет, а система сразу же создаст новый рисунок с простейшими установками. Такую же операцию создания нового рисунка выполняет кнопка  панели инструментов Standard (Стандартная), которая соответствует команде QNEW (БНОВЫЙ).

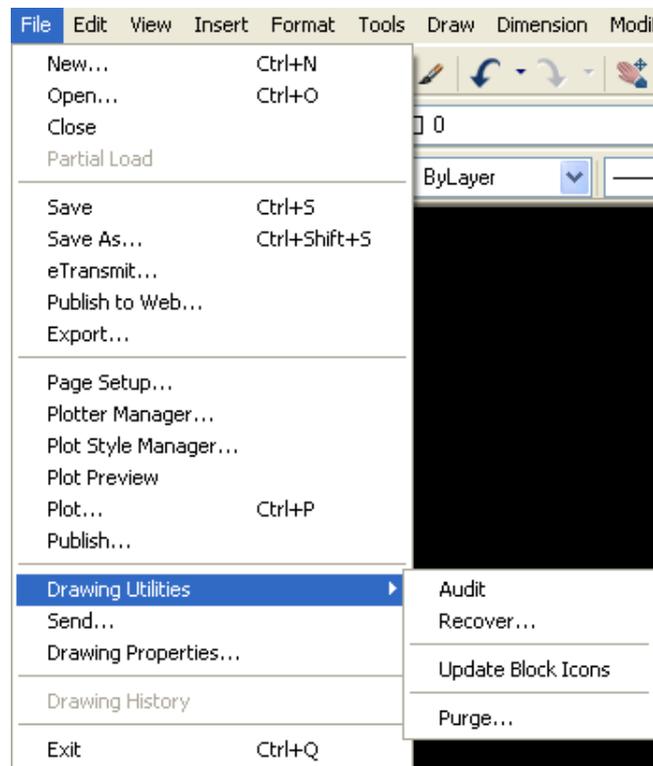


Рисунок 2.10. Падающее меню File

Система AutoCAD 2004 поддерживает многодокументный режим, поэтому в одном сеансе работы с AutoCAD можно иметь открытыми сразу несколько рисунков. Каждый из них оформляется как отдельное окно Windows. Используя кнопки сворачивания и восстановления окон (элемент, обозначенный номером 14 на рис. 2.4), которые находятся в правой верхней части, вы можете расположить рисунки так, как вам будет удобно. В этом вам помогут также команды операций над окнами, включенные в падающее меню Window

(Окно) (рисунок 2.11):

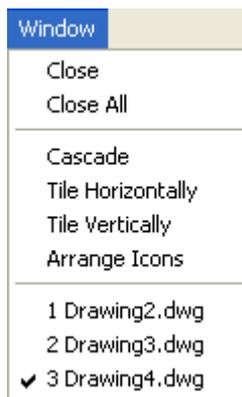


Рисунок 2.11 – Падающее меню Window

На рисунке 2.12 показаны четыре одновременно открытых рисунка, размещенные согласно пункту Arrange Icons (Упорядочить значки) меню Window (Окно).

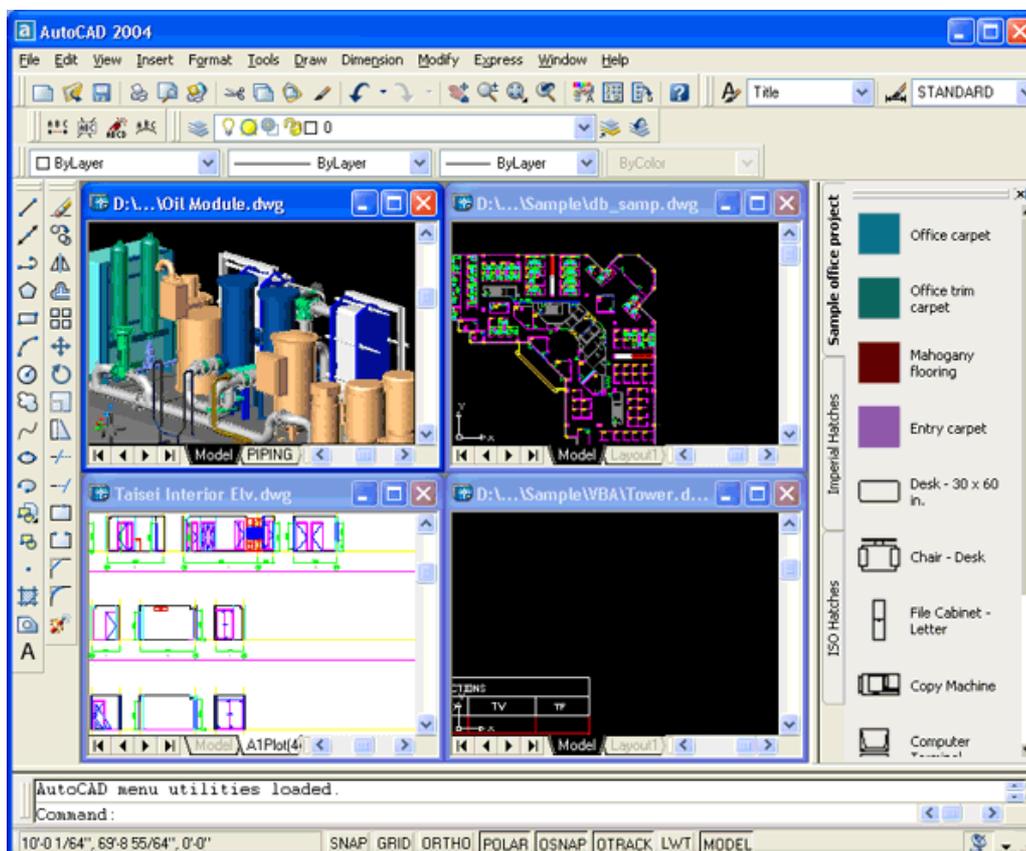


Рисунок 2.12 – Пример четырех одновременно открытых рисунков

AutoCAD 2004 имеет DWG-формат, отличный от формата систем AutoCAD 2000, 20001 и 2002. Поэтому файлы, созданные системой AutoCAD 2004, не могут открываться предыдущими версиями системы.

В рассматриваемой нами версии AutoCAD возможен экспорт файлов в следующие форматы: WMF, SAT, STL, EPS, DXX, BMP, 3DS.

## 2.5 Единицы измерения

AutoCAD может работать в двух линейных единицах измерения: в миллиметрах и дюймах и различных угловых единицах, причем единицы в процессе работы могут меняться. В разделе Default Settings (Единицы по умолчанию) окна Startup (Начало работы) включение переключателя Metric (Метрические) означает, что в качестве единиц измерения при создании рисунка выбраны миллиметры. Переход к подробной установке единиц осуществляется в окне Startup (Начало работы) или в окне Create New Drawing (Создание нового рисунка) с помощью кнопки , которая переводит начальное окно в режим вызова программы-мастера создания нового чертежа.

Мастер начальных установок может работать в одном из двух вариантов дальнейшей работы:

- Quick Setup (Быстрая подготовка);
- Advanced Setup (Детальная подготовка).

В режиме быстрой подготовки будут запрошены только тип линейных единиц и размеры зоны рисования. Более подробная настройка выполняется в режиме детальной подготовки, которым мы и продолжаем дальнейшее освоение системы AutoCAD. Выберите мышью режим Advanced Setup (Детальная подготовка) и нажмите кнопку ОК. У вас откроется первая страница окна Advanced Setup (Детальная подготовка) (рисунок 2.13).

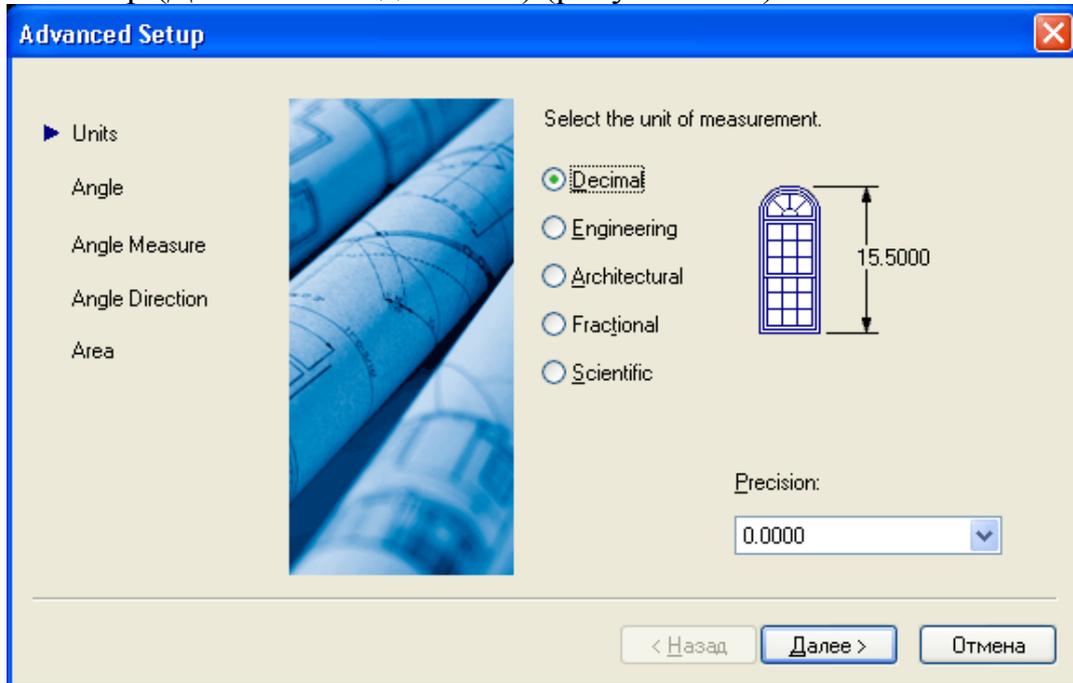


Рисунок 2.13 – Страница Units окна Advanced Setup

Выберите десятичные единицы (включите переключатель Decimal (Десятичные)). Рисунок справа иллюстрирует те единицы, которые вы выбрали. За-

тем с помощью раскрывающегося списка Precision (Точность) установите точность, с которой вы будете оперировать с вещественными числами (щелкните левой кнопкой мыши, поместив ее указатель на треугольнике раскрывающегося списка, и отметьте подходящий для вас вариант; по умолчанию это четыре знака после десятичной точки), и нажмите на кнопку Далее (Next).

На рис. 2.13 и других рисунках некоторые стандартные кнопки имеют русские названия, поскольку эти названия зависят от языка операционной системы, а не от языка системы AutoCAD. Поэтому, например, на рис. 2.13 кнопки Back, Next и Cancel приведены с наименованиями Назад, Далее и Отмена.

Появится очередная страница для задания единиц и точности углов (рисунок 2.14).

Выберите Decimal Degrees (Десятичные градусы). Рисунок справа опять покажет иллюстрацию к избранному вами способу. Установите точность измерения углов (обычно это округление до целого) и нажмите кнопку Далее (Next). Переходим к следующей странице (рисунок 2.15). В этом окне надо выбрать то направление, которое будет соответствовать нулевому углу (обычно это East (Восток)), и снова нажать кнопку Далее (Next). Откроется очередная страница настройки, задающая ориентацию отсчета углов (рисунок 2.16).

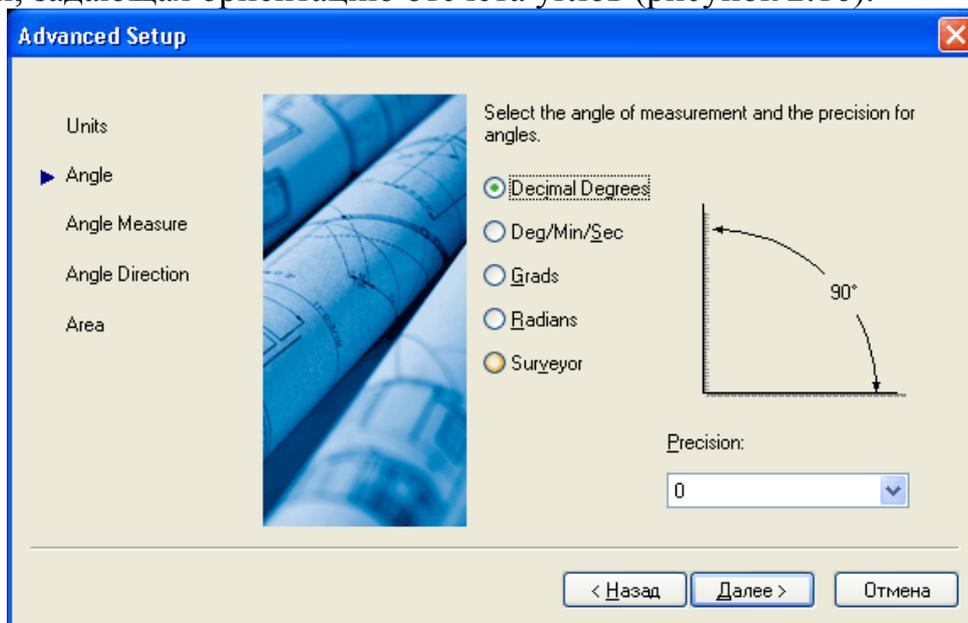


Рисунок 2.14 – Страница Angle окна Advanced Setup

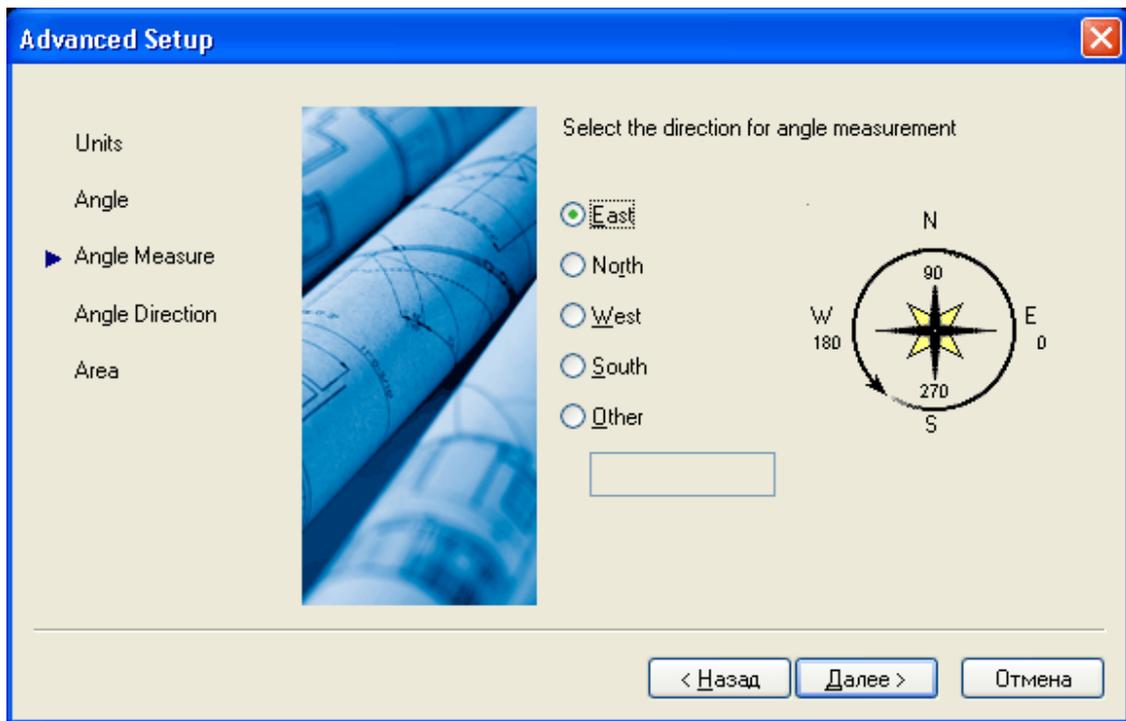


Рисунок 2.15 – Страница Angle Measure окна Advanced Setup

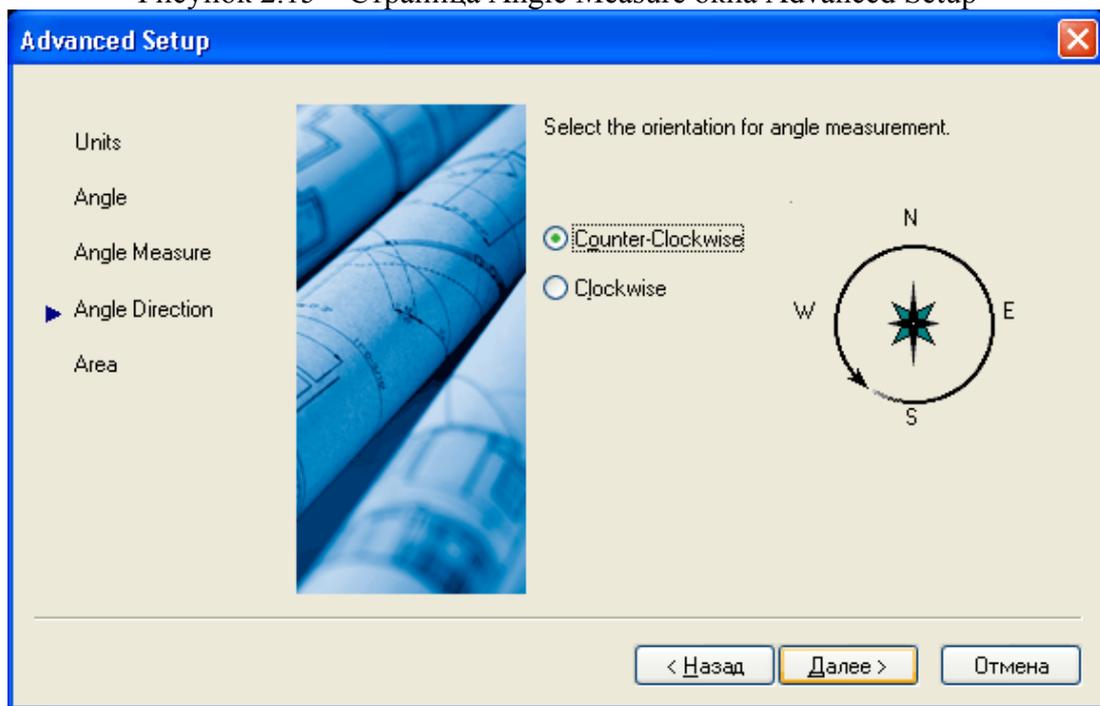


Рисунок 2.16 – Страница Angle Direction окна Advanced Setup

Здесь нужно определиться с направлением отсчета углов, т. е. отсчета по часовой стрелке или против нее. Обычный вариант — это Counter-Clockwise (Против часовой стрелки). Нажмите кнопку Далее (Next), в результате чего откроется заключительная страница, устанавливающая размеры области рисования (рисунок 2.17).

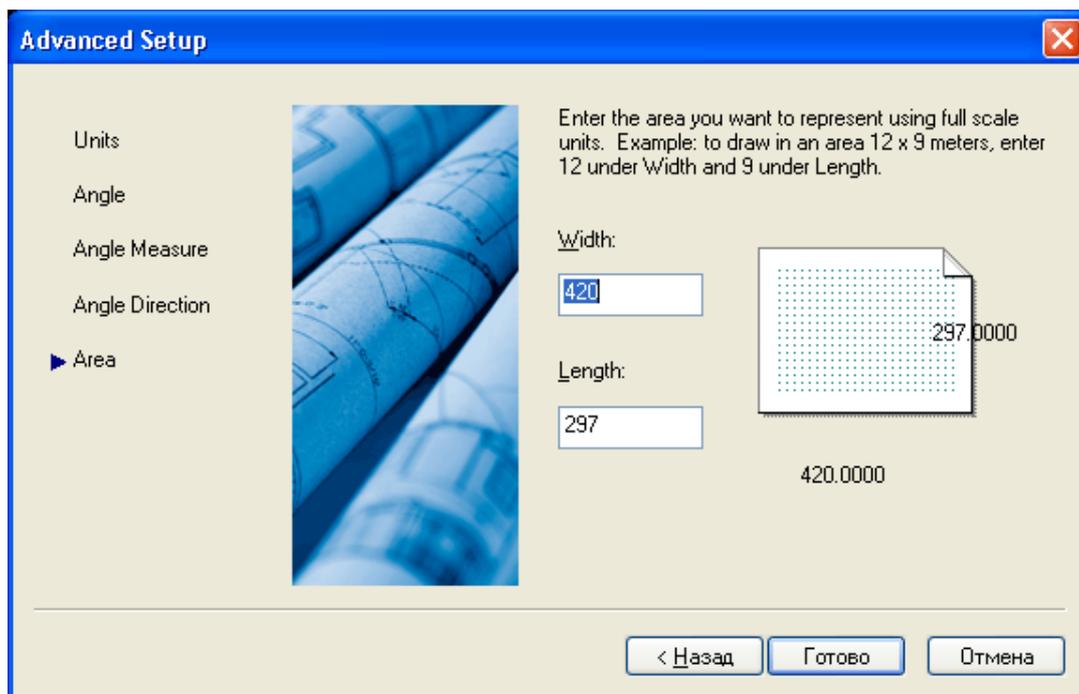


Рисунок 2.17 – Страница Area окна Advanced Setup

В новом окне вам надо определить начальные размеры зоны рисования, которая называется зоной лимитов. Как правило, левый нижний угол этой прямоугольной зоны имеет нулевые координаты как по горизонтальной, так и по вертикальной осям, а верхний правый угол определяется пользователем. По умолчанию это зона формата А3 (420 мм х 297 мм), но пользователь может выбрать и другие размеры, пользуясь двумя специальными полями: Width (Ширина) и Length (Длина) (для активизации поля нужно щелкнуть левой кнопкой мыши по этому полю, в котором после этого появится мерцающий вертикальный курсор, и ввести или отредактировать значение размера). После установки размеров остается нажать на кнопку Готово (Finish). Ваш чертеж после этого будет настроен в соответствии с выбранными вами линейными и угловыми единицами измерения. В левом нижнем углу графического экрана обычно изображается пиктограмма осей координат. Ось X экрана направлена вдоль горизонтальной кромки экрана, ось Y— вдоль вертикальной кромки. Основная система координат, в которой вы по умолчанию начинаете работу, называется мировой. Ось Z системы AutoCAD направлена от плоскости экрана к вам (конечно, это важно, только если вы работаете не на плоскости, а в пространстве). При движении указателя мыши по графическому экрану счетчик координат, если он не отключен функциональной клавишей <F6>, считывает текущие координаты и выводит их в левом нижнем углу в статусной строке. Хотя заданы размеры зоны лимитов, вы не обязаны строго придерживаться этих ограничений. Выход за пределы лимитов допустим (а когда вы с помощью AutoCAD создаете двумерную или трехмерную модель большого объекта, то тогда вам обязательно придется выйти за границы лимитов).

## 2.6 Средства организации чертежа

*Задание координат в AutoCAD.* В AutoCAD ввод координат возможен в виде *абсолютных* и *относительных* координат. Ввод абсолютных координат возможен в следующих форматах: прямоугольных (декартовых) координат (X, Y); полярных координат  $r < A$ , где  $r$  — радиус, а  $A$  — угол от предыдущей точки, заданный в градусах против часовой стрелки. Относительные координаты задают смещение от последней введенной точки. При вводе точек в относительных координатах можно использовать любой формат записи в абсолютных координатах: @ds, dy — для прямоугольных; @r<A — для полярных.

В трехмерном пространстве вместо полярных координат двумерного пространства используются *цилиндрические* (точка определяется расстоянием от начала координат вдоль направления, заданного углом относительно оси X, и значением Z вдоль перпендикуляра к плоскости XY) и *сферические* (точка определяется расстоянием от начала координат вдоль направления, заданного углом относительно оси X, и углом относительно плоскости XY) координаты.

После первой загрузки системы в левом нижнем углу появляется пиктограмма *WCS* (*World Coordinate System* – мировая система координат) (рисунок 2.18). Она направлена следующим образом: ось OX — слева направо, ось OY — снизу вверх, ось OZ — перпендикулярно экрану.

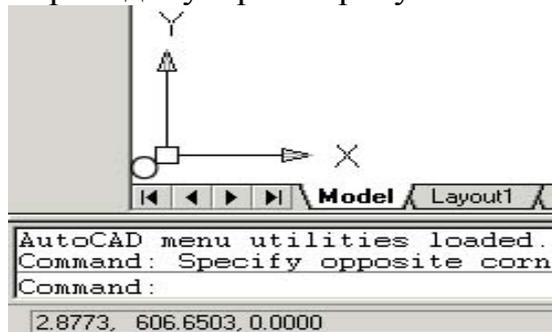


Рисунок 2.18 – Пиктограмма системы координат

Для удобства работы может быть определена *UCS* (*User Coordinate System* – пользовательская система координат), которую можно сдвинуть и / или повернуть под любым углом относительно *WCS*. Таких систем координат одновременно может существовать несколько, и в любой момент возможен переход из одной в другую. Переключение между режимами отображения значений координат – *Control-D*.

В AutoCAD имеется возможность установить *полярный* (POLAR) и *ортогональный* (ORTHO) режимы работы. *Ортогональный* режим является частным случаем *полярного*, при его установке линии направлены вдоль осей координат под углом 0, 90, 180 или 270°. Это бывает удобно использовать при редактировании, создании новых видов, параллельном переносе и т.п. Если значения координат точки вводятся с клавиатуры, то они получают более высокий приоритет и воспринимаются системой независимо от установки этого режима.

Существует два вида привязки координат: *шаговая привязка* (SNAP) –

привязка координат к узлам невидимой сетки; **объектная привязка (OSNAP)** – привязка координат к различным точкам уже созданных объектов. Переход между активными режимами привязки осуществляется с помощью клавиши **Tab**. Невидимую сетку можно сделать видимой при помощи команды **GRID**. Шаг видимой на экране вспомогательной сетки не обязательно должен совпасть с сеткой шаговой привязки.

Объектная привязка (**OSNAP**) позволяет задавать новые точки относительно характерных точек уже существующих геометрических объектов.

Управление объектной привязкой осуществляется из диалогового окна **Drafting Settings**, закладка Object Snap, которое вызывается одним из следующих способов: используя меню *Tools => Drafting Settings*; с помощью мыши подвести курсор на кнопку **OSNAP** или **OTRACK**, находящуюся в статусной строке, и по нажатии правой кнопки выбрать *Settings*, используя контекстное меню, которое вызывается щелчком правой кнопки мыши при нажатой клавише **Shift**.

В AutoCAD возможно использование многослойной организации чертежа. Каждый слой имеет свое **имя** (имя слоя может содержать до 255 символов) и характеризуется **цветом, типом и толщиной линий**, которые устанавливаются для **всех объектов, принадлежащих слою**. Слои используются для размещения на них отдельных объектов чертежа. Так как слои можно включать / выключать, то тем самым можно легко управлять содержимым чертежа. Созданием, удалением и редактированием свойств слоев управляет **Layer Properties Manager** (Диспетчер свойств слоев) (рисунок 2.19).

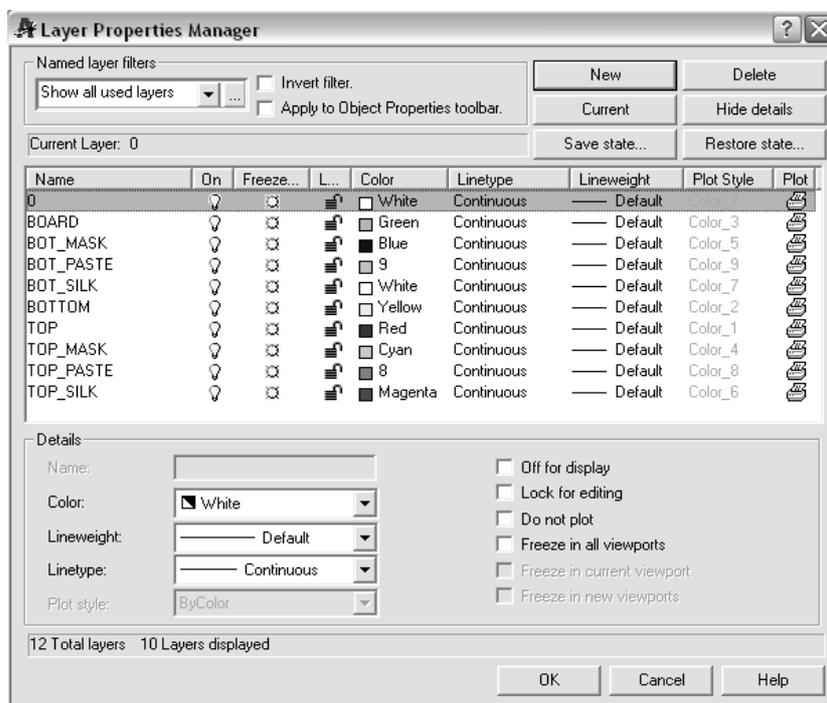


Рисунок 2.19 – Диспетчер свойств слоев (**Layer Properties Manager**)

У слоев различают следующие свойства: состояние включения / выключения; состояние замороженности на всех видовых экранах; со-

стояние блокировки; цвет графических объектов; тип линий графических объектов; вес линий (толщина в мм); признак печатаемости / непечатаемости.

Включение / выключение слоя производится в том случае, если необходимо временно убрать с чертежа все объекты, содержащиеся на данном слое.

Признак замороженности на всех видовых экранах применяется при оформлении бумажных чертежей.

Блокировка слоя применяется для того, чтобы защитить объекты, принадлежащие этому слою, от случайного удаления. При попытке удалить объект выдается сообщение о том, что слой заблокирован и удаление невозможно.

Цвет назначается обычно для того, чтобы различать линии разных слоев. При печати такого чертежа, если не пользоваться специальными установками печати, цвет будет выводиться на печать, что недопустимо по ЕСКД.

Вес линий в мм характеризует толщину графических объектов при выводе на печать. При создании новых слоев в поле «Вес линий» может стоять значение «Обычный». Это значит, что толщина равна 0,25 мм.

Печатаемость / непечатаемость слоя применяется в случае, если данный слой содержит служебную информацию (например видовые экраны), которая не должна присутствовать на печати.

Информация о текущем слое представлена на панели **Object Properties** (Свойства объектов) (рисунок 2.20). Пиктограмма 1 служит для вызова «Диспетчера свойств слоев» и запускает команду возврата к предыдущему состоянию слоев. Справа от панели «Слои» (2) находится панель «Свойства», которая содержит три раскрывающихся списка: «Цвет» объекта (3), «Тип линии» объекта (4) и «Толщина линии» объекта (5).

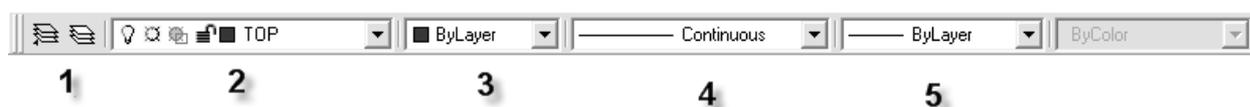


Рисунок 2.20 – Вид инструментальной панели **Object Properties** (Свойства объектов)

В раскрывающемся списке 2 представлен текущий слой (если не выбрано ни одного объекта). Если в данный момент выбраны объекты, содержащиеся на другом слое, то это поле покажет тот слой, которому они принадлежат. Если выбранные объекты принадлежат двум и более слоям, то список 2 будет чистым.

## 2.7 Создание и оформление графических объектов для двухмерного моделирования

**Примитив** – заранее определенный основной геометрический элемент, при помощи которого строятся более сложные модели.

Примитивы могут быть *простыми* и *сложными*. К простым примитивам относятся следующие объекты: точка, отрезок, круг (окружность), дуга, прямая, луч, эллипс, сплайн, однострочный текст. К сложным примитивам относятся: полилиния, мультилиния, мультитекст (многострочный текст), размер, выноска,

допуск, штриховка, вхождение блока или внешней ссылки, атрибут, растровое изображение.

**Общими свойствами**, которыми обладают все примитивы, являются *принадлежность к слою*, *цвет* и *тип линии*. Многие примитивы обладают также *толщиной*.

Отдельные примитивы (текст, блок) имеют **специальные свойства**, такие как *гарнитура*, *угол наклона*, *точка вставки* и т. д.

**Принадлежность к слою** обеспечивает размещение примитивов только на этом слое. Свойства создаваемых в нем примитивов можно определить в соответствии со свойствами слоя.

**Цвет линии** кодируется числом от 1 до 255. Использование цвета улучшает вид чертежа на экране монитора и полезно при его выводе на плоттер.

**Тип линии** – это специальная последовательность чередующихся линейных сегментов и пробелов. Начертание и назначение линий на чертежах всех отраслей промышленности и строительства устанавливается стандартом ГОСТ 2.303-81.

**Толщина линий** на чертеже устанавливается по отношению к толщине основной линии. Используя цвет, тип и толщину линий, можно выделить на чертеже важные детали, последние модификации или визуально отметить взаимосвязь отдельных фрагментов чертежа.

Операции построения большей части примитивов могут быть выполнены с помощью кнопок панели инструментов **Draw** (Рисование) (рисунок 2.21). Создавать примитивы можно также с помощью пунктов падающего меню *Draw*.



Рисунок 2.21 – Панель инструментов **Draw** (Рисование)

**Рисование отрезков.** Введите на клавиатуре **LINE** и нажмите клавишу <Enter> (любой ввод с клавиатуры должен завершаться этой клавишей). Тем самым будет вызвана команда **LINE** (ОТРЕЗОК). Данную команду можно вызвать также, воспользовавшись мышью, либо из падающего меню *Draw* с помощью команды **LINE**, либо из панели инструментов **Draw**, нажав кнопку . В завершение команды **LINE** необходимо нажать клавишу <Enter> или поместить указатель мыши внутрь графического экрана и нажать правую кнопку мыши. При этом на месте, где находился конец указателя, появится контекстное меню, содержание которого зависит от выполняемой в данный момент команды.

Одной командой **LINE** можно нарисовать один или несколько последовательно расположенных отрезков. Признаком окончания команды является нажатие клавиши <Enter> или ввод опции замыкания – символа **C**. Прерывание команды с помощью клавиши <Esc> или пункта **Cancel** контекстного меню также завершит эту команду. По любому из построенных отрезков можно с помощью команды **LIST** (СПИСОК) получить справочную информацию: координаты концов, угол наклона и т.п.

**Способы ввода координат точек.** В AutoCAD возможен ввод координат

точки с клавиатуры (*первый способ*) и с помощью мыши (*второй способ*). При вводе координат с клавиатуры запятая является разделителем между абсциссой и ординатой, а точка используется как разделитель между целой и дробной частью числа. *Третий способ* ввода точек – это относительный ввод с клавиатуры в декартовых или полярных координатах, например: @50,25. *Четвертый способ* ввода точек – это указание с помощью функций объектной привязки. Доступ к функциям объектной привязки осуществляется либо через панель **Object Snap** (Объектная привязка) (рисунок 2.22), либо через контекстное меню.



Рисунок 2.22 – Панель **Object Snap** (Объектная привязка)

В этой панели собраны следующие кнопки:

-  – использование отслеживания с помощью промежуточной точки;
-  – смещение от другой (вспомогательной) точки;
-  – конечная точка;
-  – средняя точка;
-  – точка пересечения двух объектов или их продолжений;
-  – точка мнимого пересечения двух объектов или их продолжений;
-  – точка продолжения;
-  – центр дуги, окружности или эллипса;
-  – точка квадранта дуги, окружности или эллипса (это точки, расположенные на 0, 90, 180 и 270 °);
-  – точка касания;
-  – перпендикулярно объекту;
-  – параллельно объекту;
-  – точка вставки текста, блока, внешней ссылки;
-  – узловая точка;
-  – ближайшая к объекту точка;
-  – без использования объектной привязки;
-  – настройка постоянных режимов привязки.

**Примитив точка.** Для построения точки используется команда **POINT** (ТОЧКА), которая, помимо набора на клавиатуре, может быть вызвана из панели **Draw** (Рисование) с помощью кнопки  или из падающего меню, если применить команду **Draw | Point | Single Point** (Рисование | Точка | Одиночная). Построенные заранее точки могут затем быть использованы другими командами.

Если системные переменные  $PDMODE = 0$  и  $PDSIZE = 0$ , то точка отображается в виде одного пиксела (т.е. одной точки экрана), что чаще всего неудобно. Поэтому в падающем меню *Format* (Формат) есть пункт *Point Style* (Отображение точек), который вызывает диалоговое окно **Point Style** (Отображение точек), в котором выбирается условное обозначение точки.

Точки могут быть использованы для построений новых объектов с помощью функции объектной привязки *Node* (Узел), при этом сами точки очень часто называются узловыми.

Например, пусть необходимо построить отрезок, у которого концы являются ранее созданными узловыми точками. Для этого вызовем команду **LINE** (ОТРЕЗОК) и на запрос *Specify first point* (Первая точка) с помощью левой кнопки мыши выберем кнопку  (**SNAP TO NODE** (Привязать к узлу)) панели **Object Snap** (Объектная привязка). Затем устанавливаем указатель мыши на первую узловую точку (обнаружив ее, AutoCAD обводит точку кружком). После этого опять обращаемся к функции объектной привязки *Node* (Узел) и указываем вторую узловую точку. Отрезок между заданными узлами построен.

*Луч* – это примитив, бесконечный в одну сторону и начинающийся в некоторой точке. Для его построения служит команда **RAY** (ЛУЧ). Команда может быть введена с клавиатуры или вызвана с помощью пункта *Ray* падающего меню *Draw* (Рисование). Применяется для дополнительных построений.

*Прямая*, в отличие от отрезков и лучей, – это бесконечные в обе стороны линии. Для их построения используется команда **XLINE** (ПРЯМАЯ), которая может быть введена с клавиатуры, из падающего меню *Draw* (Рисование) или кнопкой . Также применяется для дополнительных построений.

*Окружность*. Рисование окружностей (кругов) выполняется командой **CIRCLE** (КРУГ). Команду можно вызвать из панели **Draw** (Рисование) кнопкой  или из падающего меню *Draw* (Рисование).

*Дуга*. Для построения дуги используется команда **ARC** (ДУГА). Команда может быть введена с клавиатуры, вызвана из панели инструментов **Draw** (Рисование) с помощью кнопки  или из падающего меню *Draw* (Рисование), в котором подменю *Arc* (Дуга) имеет одиннадцать пунктов для уточнения способов построения дуги.

*Полилиния* – это сложный примитив, состоящий из одного или нескольких связанных между собой прямолинейных и дуговых сегментов. Полилиния обрабатывается как единое целое (например при редактировании или удалении). Для рисования полилинии служит команда **PLINE** (ПОЛИЛИНИЯ), которая, помимо ввода с клавиатуры, может быть вызвана с помощью кнопки  или пункта *Polyline* (Полилиния) падающего меню *Draw* (Рисование).

Полилиния – один из немногих объектов, которые могут иметь ненулевую ширину. Ширина, заданная для предыдущей полилинии, запоминается и предлагается в качестве ширины по умолчанию для следующей полилинии. Поэтому AutoCAD информирует вас о том, с какой шириной система будет строить

новую полилинию. В любой момент можно от режима рисования прямолинейных сегментов перейти к режиму рисования дуговых сегментов и наоборот. Также в любой момент можно задать новую ширину или полуширину для следующего сегмента полилинии.

Особенности полилинии по сравнению с простыми примитивами:

- полилиния является единым объектом, что удобно для операций удаления или редактирования (например построения параллельной линии);
- полилиния удобна для рисования жирных линий чертежа;
- переменная ширина сегментов полилинии может быть использована для графических эффектов (построения стрелок и т. п.).

*Полилинии специального вида.* В системе AutoCAD есть несколько команд рисования таких объектов, как прямоугольники, правильные многоугольники, кольца и линии правки, каждый из которых на самом деле является полилинией. Вычерчивание прямоугольников осуществляет команда **RECTANG** (ПРЯМО-УГ). Команда может быть введена с клавиатуры или вызвана с помощью кнопки  панели **Draw** (Рисование) или с помощью пункта *Rectangle* (Прямоугольник) падающего меню *Draw* (Рисование).

Команда **POLYGON** (МН-УГОЛ) рисует правильный многоугольник либо по конечным точкам одной стороны, либо по точке центра и радиусу вписанной или описанной окружности. Команда может быть вызвана с помощью кнопки  панели **Draw** (Рисование) или с помощью пункта *Polygon* (Многоугольник) падающего меню *Draw* (Рисование).

Кольцо рисуется командой **DONUT** (КОЛЬЦО) и представляется полилинией с шириной, подобранной по внутреннему и внешнему диаметрам кольца. Команда, помимо ввода с клавиатуры, может быть вызвана с помощью пункта *Donut* (Кольцо) падающего меню *Draw* (Рисование).

Новая команда, появившаяся в системе AutoCAD 2004, - **REVCLOUD**. Эта команда рисует замкнутую полилинию в форме облака.

*Мультилиния* – это объект, состоящий из пучка ломаных, параллельных друг другу линий. Количество линий, входящих в мультилинию, составляет от 2 до 16. Мультилиния может обладать дополнительными свойствами, к которым относятся промежуточные стыки, торцы, скругления и заливка. Построение мультилиний выполняется командой **MLINE** (МЛИНИЯ). Команда может быть введена с клавиатуры или вызвана с помощью пункта *Multiline* (Мультилиния) падающего меню *Draw* (Рисование).

*Эллипс* – это геометрическое место точек, сумма расстояний до которых от двух фиксированных точек (фокусов) постоянна. Создание эллипсов и эллиптических дуг выполняется в системе AutoCAD с помощью команды **ELLIPSE** (ЭЛЛИПС), которая, помимо ввода с клавиатуры, может быть вызвана кнопкой  панели **Draw** (Рисование) или пунктом падающего меню *Draw* | *Ellipse* | *Axis, End* (Рисование | Эллипс | Ось, Конец).

*Сплайн.* Команда **SPLINE** (СПЛАЙН) позволяет чертить сплайн – гладкую линию, которая точно проходит через заданные точки или отклоняется от них в

рамках допуска и может удовлетворять условиям касания в начальной, конечной или обеих точках. Команда, помимо ввода с клавиатуры, может быть вызвана с помощью кнопки  панели **Draw** (Рисование) или пункта *Spline* (Сплайн) падающего меню **Draw** (Рисование). Сплайны могут применяться при оформлении чертежа в качестве линий разрыва, определения границ местных разрезов и сечений и т.п.

*Надписи* могут быть созданы с помощью команд **ТЕХТ** (ТЕКСТ) или **МТЕХТ** (МТЕКСТ). В первом случае создается однострочный текст, во втором – мультитекст (многострочный текст). У команды **ТЕХТ** есть синоним – команда **ДТЕХТ** (ДТЕКСТ). Команда **ТЕКСТ**, создающая простые надписи, помимо клавиатуры, может быть вызвана кнопкой  панели инструментов **Text** (Текст) (рисунок 2.23), а также из падающего меню **Draw** (Рисование), где в подменю *Text* (Текст) есть пункт *Single Line Text* (Однострочный текст).



Рисунок 2.23 – Панель инструментов **Text** (Текст)

Если в текст нужно вставить специальные знаки (например диаметр) или получить подчеркнутые или надчеркнутые символы, то при вводе текста можно использовать следующие управляющие коды, начинающиеся с двух символов процента:

- %%nnn — вставка символа с номером nnn (nnn заменяет три цифры) в текущей таблице кодировки символов, используемой как основной в графических приложениях для Windows (например, %%193 — это символ с номером 193, т. е. "Б");
- %%o — включение / отключение надчеркивания;
- %%u — включение / отключение подчеркивания;
- %%d — вставка символа градуса "°";
- %%p — вставка символа плюс-минус "±";
- %%c — вставка символа диаметра;
- %%% — вставка символа процента "%".

Режимы надчеркивания и подчеркивания могут действовать в надписи одновременно.

Команда **МТЕХТ** (МТЕКСТ) позволяет нанести на чертеж целые абзацы достаточно длинного текста (образующийся при этом примитив называется мультитекстом, или многострочным текстом), с возможностями выравнивания и редактирования, приближающимися к возможностям таких текстовых процессоров, как *Microsoft Word*. Команде соответствуют кнопка  в панелях **Draw** (Рисование) и **Text** (Текст), а также пункт падающего меню **Draw** | *Text* | *Multiline Text* (Рисование | Текст | Многострочный текст). Команда **МТЕХТ** (МТЕКСТ) при старте информирует об имени действующего текстового стиля и просит указать первую из двух точек, определяющих границы зоны мульти-

текста по ширине. Нижний предел по высоте при этом не фиксируется и сдвигается автоматически по мере ввода текста. После указания точек раскрывается окно редактора мультитекста.

Перед вводом текста необходимо установить нужный стиль или задать внестилизовое оформление с помощью имени шрифта и размера букв, а также выбрать цвет букв. Если в процессе ввода понадобится сменить настройки (имя шрифта, цвет и т. д.), то можно выделить левой кнопкой мыши участок текста и изменить его форматирование. Имена шрифтов имеют слева пометки в виде значка  (это стандартные SHX-шрифты AutoCAD) или  (шрифты Windows типа True Type). Не все шрифты могут воспроизводить русские буквы, поэтому требуется предварительный анализ шрифтов перед их практическим использованием. Шрифты, имена которых заканчиваются на *Cyr*, всегда подходят для русского текста.

Кнопка  позволяет управлять написанием дробей и индексов. Числитель и знаменатель дроби могут либо разделяться горизонтальной чертой, либо ничем не разделяться (по типу допуска), либо разделяться наклонной чертой. Перед тем как воспользоваться кнопкой дробей, нужно выделить мышью участок текста, в котором будущие числитель и знаменатель разделены одним из символов: /, ^ или #. Затем нужно щелкнуть по указанной кнопке. Применение данной кнопки к выделенному участку текста, уже имеющему вид дроби, возвращает этот участок к неформатированному виду. Для получения верхнего или нижнего индекса можно воспользоваться средством форматирования дроби с помощью специального символа *A*.

При создании многострочного текста задается межстрочный интервал фиксированным численным значением или в долях от одинарного интервала. Величина одинарного межстрочного интервала обозначается буквой *x* и равна высоте символов, умноженной на 1,66. Поэтому значение межстрочного интервала можно ввести как число с суффиксом *x* (например, 1,5 *x*, что соответствует полуторному интервалу) или просто как число (например 2,25, что соответствует 2,25 от высоты символов). Значения должны находиться в диапазоне от 0,0833 (0,25*x*) до 1,3333 (4*x*). Сочетание правильно подобранного межстрочного интервала и позиций табуляции позволяет эффективно использовать мультитекст для заполнения таблиц (например таблицы спецификации, располагающейся на чертеже).

*Размеры.* Операции установки размеров, допусков и выносных линий (выносок) выполняются с помощью команд, которым соответствуют пункты падающего меню *Dimension* (Размеры) и кнопки панели инструментов *Dimension* (Размеры) (рисунок 2.24).



Рисунок 2.24 – Панель инструментов *Dimension* (Размеры)

По умолчанию AutoCAD 2004 и AutoCAD 2002 в отличие от предыдущих

версий все размеры создают ассоциативными, т.е. зависимыми от объектов, к которым данные размеры привязаны. Это означает, что при редактировании основного объекта будут автоматически изменяться и все связанные с ним размеры.

Команда **DIMLINEAR** предназначена для установки линейных размеров. Ей соответствуют кнопка  панели **Dimension** (Размеры) и пункт *Linear* (Линейный) падающего меню *Dimension* (Размеры).

Команда **DIMALIGNED**, которой соответствуют кнопка  панели **Dimension** (Размеры) и пункт *Aligned* (Параллельный) падающего меню *Dimension* (Размеры), позволяет проставить линейный размер параллельно выбранному отрезку или двум указанным точкам.

Команда **DIMORDINATE** позволяет строить выноски с установкой значения абсциссы или ординаты указываемой точки. Данной команде соответствуют кнопка  панели **Dimension** (Размеры) и пункт *Ordinate* (Ординатный) падающего меню *Dimension*.

Команде **DIMRADIUS** соответствуют кнопка  панели инструментов **Dimension** (Размеры) и пункт *Radius* (Радиус) падающего меню *Dimension*. Команда предназначена для простановки радиуса.

Команда **DIMDIAMETER** предназначена для нанесения диаметра. Ей соответствуют кнопка  панели инструментов **Dimension** (Размеры) и пункт *Diameter* (Диаметр) падающего меню *Dimension*.

Команда **DIMANGULAR** предназначена для простановки угловых размеров между отрезками или углового размера дуги (или части окружности). Ей соответствуют кнопка  панели инструментов **Dimension** (Размеры) и пункт *Angular* (Угловой) падающего меню *Dimension*.

Команда **QDIM** предназначена для быстрого создания группы однотипных размеров или для быстрого построения базовых размеров и размерных цепей.

Команда **DIMBASELINE** позволяет от одной и той же базовой точки построить несколько линейных размеров. Эта команда помимо ввода с клавиатуры может быть вызвана с помощью кнопки  панели инструментов или пункта *Baseline* (Базовый) падающего меню *Dimension*. Команда не запрашивает положения первой выносной линии, а сразу начинает с запроса второй. В качестве базы группы базовых размеров обычно служит предыдущий линейный размер, и его первая точка становится первой точкой для следующих линейных (базовых) размеров.

Команда **DIMCONTINUE** позволяет построить группу продолжающих друг друга линейных размеров. Эта команда помимо ввода с клавиатуры может быть вызвана с помощью кнопки  панели инструментов или пункта *Continue* (Цепь) падающего меню *Dimension* (Размеры). Команда **DIMCONTINUE** начинает сразу с запроса положения второй выносной линии. Эти выносные линии можно поочередно указать в цикле. Опция *Undo* (Отменить) отменяет предыдущий шаг этой команды, а опция *Select* (Выбрать) позволяет выбрать другой линейный размер в качестве базы для построения размерной цепи.

Команда **QLEADER** строит одноименный примитив выноски, который состоит из ломаной либо гладкой выносной линии или из нескольких сегментов, начинающихся стрелкой (или другим настраиваемым символом) и заканчивающихся одной либо несколькими строками текста или мультитекста. Команда вызывается кнопкой  панели **Dimension** (Размеры) или пунктом *Leader* (Выноска) падающего меню *Dimension* (Размеры). Окончание текста выноски – нажатие клавиши **<Enter>**.

В последних версиях системы AutoCAD выноска имеет развитый режим настройки, в который можно войти, выбрав вместо первой точки выноски пункт *Settings* (Параметры) или нажав клавишу **<Enter>**. В этом случае появляется диалоговое окно **Leader Settings** (Параметры выноски) (рисунок 2.25).

Команда **TOLERANCE** (ДОПУСК) формирует обозначение допуска в виде нескольких рядов прямоугольников (от одного до четырех). Команде соответствует кнопка  панели **Dimension** (Размеры) и пункт *Tolerance* (Допуск) падающего меню *Dimension* (Размеры). Эта команда открывает диалоговое окно **Geometric Tolerance** (Допуски формы и расположения) (рисунок 2.26). Обозначение допуска строится с помощью этого диалогового окна, четыре строки которого соответствуют четырем строкам допуска. Если создаваемое обозначение допуска должно иметь меньшее количество строк, то соответствующая строка диалогового окна не заполняется. Каждая строка диалогового окна рассчитана на максимальную длину (до 13 элементов). Поэтому если реальное обозначение допуска будет иметь меньшую длину, то ненужные элементы не заполняются, и они не включаются программой в формируемый примитив допуска.

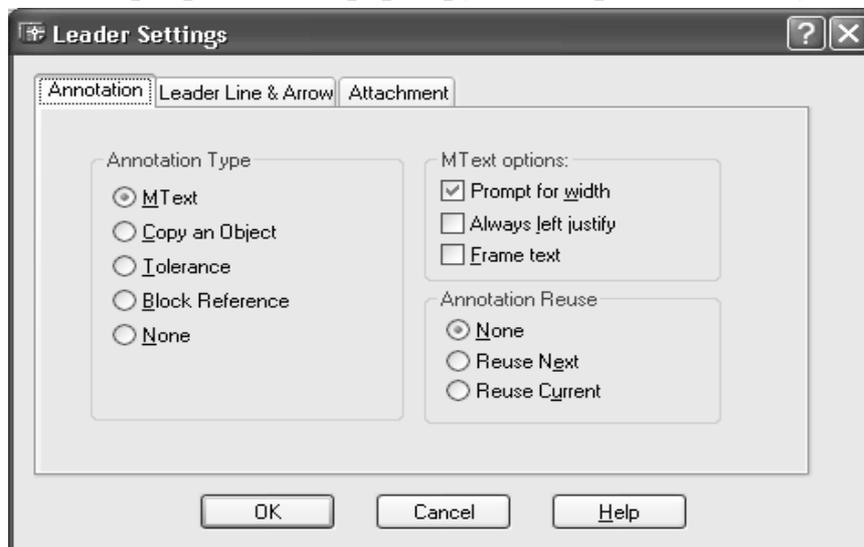


Рисунок 2.25 – Диалоговое окно **Leader Settings**, вкладка *Annotation*

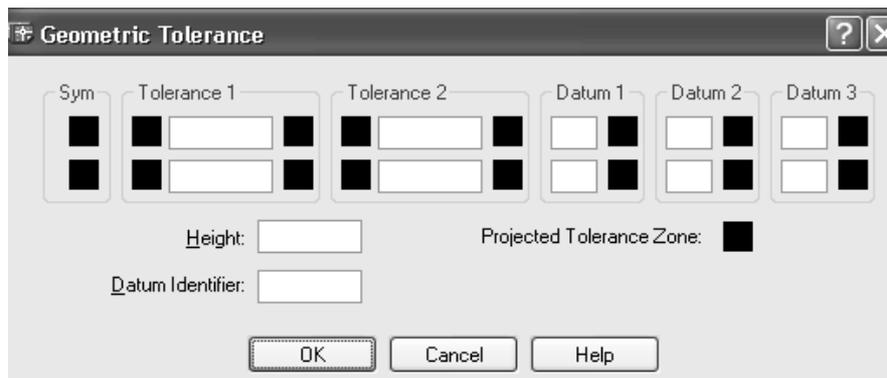


Рисунок 2.26 – Диалоговое окно **Geometric Tolerance**  
(Допуски формы и расположения)

Заполнение полей в диалоговом окне выполняется следующим образом. Черные поля являются полями выбора символов из специальных окон, а белые поля – текстовые, содержимое которых заполняет пользователь. Если щелкнуть левой кнопкой мыши по полю первой или второй строки раздела *Sym* (Симв), то раскроется окно **Symbol** (Символ) (рисунок 2.27), в котором нужно выбрать один из предлагаемых вариантов.

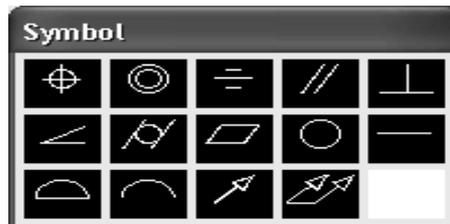


Рисунок 2.27 – Окно **Symbol** (Символ)

Если щелкнуть левой кнопкой мыши по верхнему левому или нижнему левому черному полю, являющемуся первой частью разделов *Tolerance 1* (Допуск 1) или *Tolerance 2* (Допуск 2), то в квадрате появится символ диаметра. Такой же щелчок удаляет символ диаметра, если это поле ранее уже было заполнено. Верхнее правое или нижнее правое черное поле *Tolerance 1* (Допуск 1) либо *Tolerance 2* (Допуск 2) может быть заполнено одним из символов зависимого допуска с помощью окна **Material Condition** (Зависимый допуск) (рисунок 2.28).

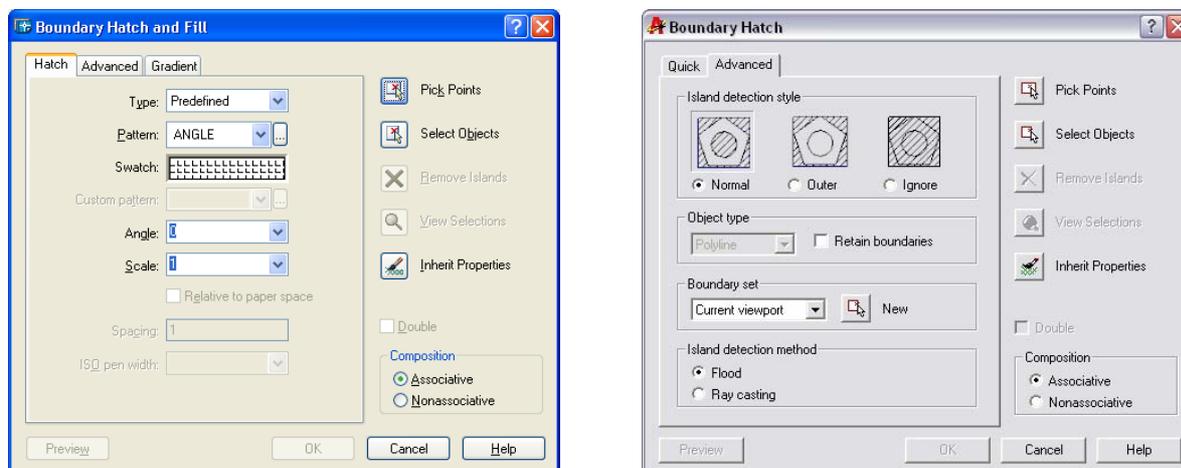
Аналогично заполняются или пропускаются остальные поля.



Рисунок 2.28 – Окно **Material Condition**  
(Зависимый допуск)

**Штриховка.** Для штрихования замкнутых областей рисунка служит команда **ВНАТЧН**, которая вызывается либо с помощью кнопки  панели **Draw** (Рисование), либо с помощью пункта *Hatch* (Штриховка) падающего меню *Draw* (Рисование). Эта команда позволяет создать штриховку или заливку области, ограниченной замкнутой линией (линиями), как путем простого указания точек внутри контура, так и путем выбора объектов. Эта команда автоматиче-

ски определяет контур и игнорирует примитивы, которые не имеют отношения к контуру. Она вызывает диалоговое окно **Boundary Hatch and Fill** (Штриховка и заливка по контуру) с тремя вкладками в левой части (рисунок 2.29).



а

б

Рисунок 2.29 – Диалоговое окно **Boundary Hatch and Fill** вкладки Hatch AutoCAD 2004 (а) и Advanced AutoCAD 2000i (б)

Диалоговое окно имеет три сменяющих друг друга вкладки в левой части. Вкладка Hatch (Штриховка) (в версиях до AutoCAD 2004 она называлась Quick (Основные)) задает параметры штриховки. Заполняемая штриховкой область задается с помощью параметров в правой части этого окна. Выбирается тип штриховки, угол наклона и масштаб штриховки относительно эталонного изображения. При увеличении масштаба расстояние между линиями штриховки увеличивается, при уменьшении масштаба — уменьшается. Та кже должны быть заданы параметры заполняемого контура.

В версии системы AutoCAD 2004 впервые появилось целое множество заливок — градиентные заливки. Для работы с ними предназначена вкладка Gradient (Градиентные). В данной версии вставка нужной штриховки или заливки может быть выполнена простым перетаскиванием образца внутрь той области, которую необходимо заполнить.

*Редкие примитивы.* К редким примитивам могут быть отнесены полосы и фигуры.

Команда **TRACE** (ПОЛОСА) вводится только с клавиатуры и предназначена для построения полос — ломаных линий с постоянной шириной.

Команда **SOLID** (ФИГУРА) строит последовательно расположенные закрашенные четырехугольники или треугольники. Вершины фигур задаются точками в диалоговом режиме. Фигуры, построенные с помощью одной команды **SOLID**, не образуют единого объекта.

*Получение справок.* В процессе работы очень полезными оказываются команды получения справочной информации о создаваемых объектах. На рисунке 2.30 показана панель инструментов **Inquiry** (Справки), содержащая кнопки следующих операций (в порядке их расположения в панели слева направо):

**Distance** (Расстояние); **Area** (Площадь); **Region | Mass Properties** (Область | Масса); **List** (Список); **Locate Point** (Координаты).



Рисунок 2.30 – Панель *Inquiry*  
(Справки)

К этим операциям можно также обратиться с помощью подменю *Inquiry* падающего меню *Tools* (Сервис).

## 2.8 Редактирование графических объектов

Наиболее быстрым способом редактирования объектов в AutoCAD является использование меток – небольших квадратов в характерных точках объектов. У разных примитивов высвечиваются различные метки.

При активации текущей метки (нажатие на левую кнопку мыши) она должна сменить цвет на красный и система AutoCAD выдает сообщение о возможных действиях при редактировании объекта.

Например, для примитива *LINE* (отрезок) характерными точками являются конечные и средние точки.

У примитива *XLINE* (прямая) высвечиваются базовая точка и две точки на прямой на небольшом расстоянии от базовой. Если переносить базовую (среднюю) точку, то перемещается вся линия, а если переносить точки других ручек, то базовая точка остается на месте, но изменяется наклон прямой.

На окружности высвечивается пять меток: в центре и квадрантах (т.е. в крайней верхней, нижней, левой и правой точках). При попытке перемещения центральной метки окружность перемещается на новое место, а в случае редактирования окружности за любую из остальных меток окружность растягивается или сжимается, меняя радиус.

У примитива *ARC* (дуга) высвечиваются три метки: на концах и в середине. Примитивы *TEXT* (текст) и *MTEXT* (мультитекст) имеют метки в тех точках, которые характеризуют положение или выравнивание текста.

Примитив *DIMENSION* имеет метки не только у базовых точек, но и на концах выносных линий и у размерного текста. Эти метки позволяют менять положение выносной линии и размерного текста (при этом общее оформление размера как единого целого сохраняется).

*Команды общего редактирования.*

Кнопки команд общего редактирования объектов (копирование, перенос, удлинение и т. п.) расположены в панели *Modify* (Редактирование) (рисунок 2.31).



Рисунок 2.31 – Панель **Modify** (Редактирование)

Каждую из команд, рассмотренных в данном разделе, можно ввести с клавиатуры, а также вызвать с помощью падающего меню *Modify* (Редактирование). Многие команды данной группы работают либо с набором предварительно выбранных объектов, либо при отсутствии такого набора выдают запрос *Select objects* (Выберите объекты). Остальные команды запрашивают редактируемые объекты в соответствующий момент. Кнопка , соответствующая команде **ERASE** (СТЕРЕТЬ), стирает с экрана выбранные объекты и удаляет их из рисунка.

Кнопка  команды **COPY** (КОПИРОВАТЬ) копирует выбранные объекты параллельно вектору, который задается начальной и конечной точками. Если при запуске команды выбранных объектов нет, то задается вопрос о выборе объектов. После выбора объектов нужно завершить выбор либо нажатием клавиши **<Enter>**, либо щелчком правой кнопкой мыши.

Кнопка  команды **MIRROR** (ЗЕРКАЛО) позволяет зеркально отразить (симметризовать) выбранные объекты относительно оси, которая определяется двумя точками. Рекомендуется применять при включенном режиме *ORTHO*.

Если среди симметризуемых объектов есть тексты, то в некоторых условиях они тоже отражаются относительно оси, что приводит к их нечитаемости. Для того чтобы тексты при отражении не переворачивались, нужно установить для системной переменной *MIRRTXT* значение 0 (вместо другого возможного значения 1). Тогда точки привязки текста симметризируются, а сам текст останется повернутым в ту же сторону, что и исходный.

Кнопка  в панели **Draw** (Рисование) соответствует команде **OFFSET** (ПОДОБИЕ). Команду можно также вызвать из падающего меню *Modify* (Редактирование) с помощью пункта *Offset* (Подобие). Команда предназначена для рисования подобных (параллельных) линий к линейным объектам (отрезкам, лучам, прямым, полилиниям, дугам, окружностям, эллипсам и сплайнам).

Кнопка  команды **ARRAY** (МАССИВ) предназначена для создания группы копий одних и тех же объектов, причем копии располагаются по определенному закону в гнездах прямоугольного или кругового массива. Команду, кроме того, можно также вызвать из падающего меню *Modify* с помощью пункта *Array*. В любом случае команда вызывает диалоговое окно **Array**, которое в левой части имеет изменяемую область. Содержимое этой области зависит от состояния двух переключателей *Rectangular Array* (Прямоугольный массив) и *Polar Array* (Круговой массив), расположенных в верхней части окна (рисунок 2.32).

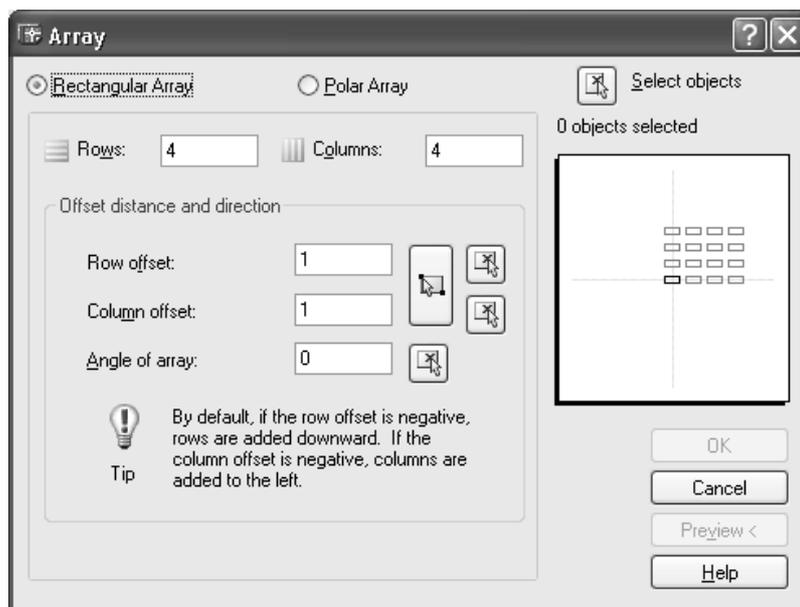


Рисунок 2.32 – Диалоговое окно **Array** с включенным переключателем *Rectangular Array*

Кнопка  соответствует команде **MOVE** (ПЕРЕНЕСТИ), которая позволяет переместить выбранные объекты параллельно вектору, заданному двумя точками. Команду можно также вызвать из падающего меню *Modify* с помощью пункта *Move* (Перенести). Запросы и действия этой команды очень похожи на запросы и действия команды **COPY**.

Кнопка  команды **ROTATE** (ПОВЕРНУТЬ) дает возможность повернуть выбранные объекты относительно базовой точки на заданный угол. Команду можно также вызвать из падающего меню *Modify* с помощью пункта *Rotate*.

Кнопка , соответствующая команде **SCALE** (МАСШТАБ), позволяет масштабировать выбранные объекты относительно базовой точки. Команду можно также вызвать из падающего меню *Modify* с помощью пункта *Scale*.

Команда **STRETCH** (РАСТЯНУТЬ), которой соответствует кнопка  панели *Modify*, предназначена для изменения формы объекта методом растяжения. Команду можно также вызвать из падающего меню *Modify* с помощью пункта *Stretch* (Растянуть). Команда обычно выполняется над полилинией (хотя она может растягивать и дуги, эллиптические дуги, сплайны и другие линейные примитивы), когда надо переместить несколько ее вершин параллельно вектору, задаваемому двумя точками, а остальные вершины полилинии надо оставить на месте и соответствующим образом преобразовать геометрию объекта. Команда **STRETCH** не применяется для набора предварительно выбранных объектов, поскольку в данном случае важно часть полилинии выбрать с помощью секущей рамки или секущего многоугольника.

Кнопка  соответствует команде **TRIM** (ОБРЕЗАТЬ), позволяющей обрезать объект (объекты) с помощью пересекающих его (их) других объектов или удлинить его (их) до нужного объекта. Команду можно также вызвать из падающего меню *Modify* с помощью пункта *Trim*. При обрезке может происхо-

дить преобразование типов примитивов. Например, окружность может стать дугой, луч – отрезком, прямая – лучом или отрезком.

Команда **EXTEND** (УДЛИНИТЬ), которой соответствует кнопка , позволяет выбрать набор "граничных кромок", а затем указать объекты, которые удлинятся до этих кромок. Команду можно также вызвать из падающего меню *Modify* с помощью пункта *Extend*. Последовательность указания объектов очень важна, так как системе нужно различать граничные и удлиняемые объекты.

Команда **BREAK** (РАЗОРВАТЬ) предназначена для разрыва объекта в двух указываемых точках. Команду можно вызвать из падающего меню *Modify* помощью пункта *Break*. Кроме того, ей в панели инструментов *Modify* соответствуют две кнопки. Кнопка  вызывает команду **BREAK** в самом общем варианте. Кнопка  **BREAK AT POINT** (Разорвать в точке) позволяет воспользоваться вариантом команды **BREAK** в том случае, когда первая и вторая точки разрыва совпадают, но точка указания объекта не является точкой разрыва.

Кнопка  вызывает команду **CHAMFER** (ФАСКА), которая выполняет операцию подрезки двух пересекающихся прямолинейных сегментов (отрезков, лучей, прямых) на заданных расстояниях от точки их пересечения (снятие фаски), строя при этом новый отрезок, соединяющий точки подрезки. Команду, кроме того, можно вызвать из падающего меню *Modify* с помощью пункта *Chamfer*. Команда выполняется как над пересекающимися, так и над непересекающимися, но непараллельными отрезками (при этом отрезки сначала удлиняются до пересечения).

Команда **FILLET** (СОПРЯЖЕНИЕ) сопрягает объекты (например отрезки, дуги и окружности) дугой заданного радиуса. Ей соответствует кнопка  панели *Modify* и пункт *Fillet* падающего меню *Modify*. Команда по своим режимам аналогична команде **CHAMFER** (ФАСКА).

Кнопка  панели *Modify*, соответствующая команде **EXPLODE** (РАСЧЛЕНИТЬ), расчленяет на более простые объекты полилинии, вхождения блоков, размеры и другие сложные объекты.

Операции редактирования текстов собраны в панели инструментов *Text* (Текст) (см. рис. 8). Для редактирования надписей предусмотрены кнопка  панели и пункт падающего меню **Modify | Object | Text | Edit** (Редактирование | Объект | Текст | Редактировать). Они вызывают команду **DDEDIT**. Другой, более простой способ вызова этой команды – двойной щелчок по однострочному тексту (при этом в рисунке не должны быть выделены другие объекты). Команда **DDEDIT** предназначена не только для редактирования текста, но и размерного текста, допусков и определений атрибутов.

Редактирование мультитекста выполняется по тем же правилам, что и создание мультитекста. Кроме того, можно выделить любой участок текста и вызвать правой кнопкой мыши контекстное меню.

Кнопка  панели *Text* вызывает команду **FIND** (НАЙТИ), которая выполняет операции поиска и замены в текстовых примитивах всего чертежа или

выделенного набора объектов. Кнопка  соответствует команде **STYLE** (СТИЛЬ), создающей стили текста. Кнопка  вызывает команду **SCALETEXT**, которая позволяет масштабировать текст. Команду можно также вызвать с помощью пункта падающего меню **Modify | Object | Text | Scale** (Редактирование | Объект | Текст | Масштаб).

Команда **JUSTIFYTEXT** (ВЫРТЕКСТ), которой соответствует кнопка  позволяет, не меняя внешнего вида надписи, изменить способ ее выравнивания (например, надпись привязана за нижнюю правую точку, а вы хотите, чтобы надпись была пересоздана с опцией *Fit* (По ширине)). Команду можно также вызвать с помощью пункта падающего меню **Modify | Object | Text | Justify** (Редактирование | Объект | Текст | Выравнивание). Изменение способа выравнивания позволяет в дальнейшем редактировать надпись (даже с помощью меток), но с сохранением новых свойств.

Кнопка  соответствует команде **SPACETRANS**, которая предназначена для пересчета длин между пространствами модели и листа.

*Редактирование размеров.* Команда **DIMEDIT** позволяет поменять размерный текст и его местоположение, а также наклонить выносные линии. Команде соответствуют кнопка  панели **Dimension** (Размеры) и пункты подменю *Align Text* (Размерный текст) падающего меню *Dimension*.

Команде **DIMSTYLE** соответствует кнопка , которая входит как в панель **Dimension** (Размеры), так и в новую панель **Styles** (Стили). Этой же команде соответствуют пункт *Style* (Стиль) падающего меню *Dimension* и пункт *Dimension Style* (Размерные стили) падающего меню *Format*. Команда позволяет создавать и редактировать размерные стили (совокупность установок, описывающих форму размерных примитивов). Эту кнопку удобно использовать еще и для внесения отдельных изменений в оформление размеров, не редактируя размерного стиля и не создавая нового.

Размеры можно корректировать с использованием диалогового окна **Dimension Style Manager** (Диспетчер размерных стилей), открываемое командой **DIMSTYLE**. Все установки оформления размеров записываются в так называемые размерные переменные, которые сохраняются вместе с рисунком. Посмотреть их действующие значения можно, щелкнув мышью по кнопке **COMPARE** (Сравнить).

*Блоки.* Важным инструментом автоматизации процесса разработки чертежей и моделей является использование блоков и внешних ссылок. Блок – это сложный именованный объект, для которого создается описание, включающее любое количество примитивов текущего рисунка. Блок имеет базовую точку; может применяться для вставки в любое место чертежа, причем в процессе вставки возможен его поворот и масштабирование с различными коэффициентами по разным осям. Блок может содержать атрибуты – переменные надписи, задаваемые пользователем.

Применяются два типа блоков: *локальные* и *автономные*.

**Локальные блоки** входят в состав того чертежа, в котором они были созданы, и не могут быть перенесены в другой чертеж.

**Автономные блоки** хранятся в виде отдельного файла чертежа и могут быть использованы в других чертежах.

Для создания блока применяют команду **BLOCK** (БЛОК). Команде соответствуют кнопка  панели инструментов **Draw** (Рисование) и пункт падающего меню **Draw | Block | Make** (Рисование | Блок | Создать). Команда **BLOCK** вызывает диалоговое окно **Block Definition** (Описание блока) (рисунок 2.33).

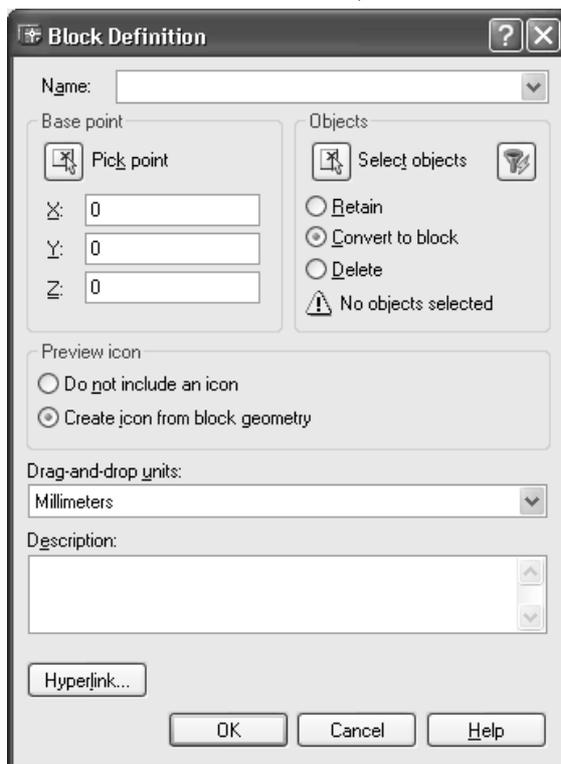


Рисунок 2.33 – Диалоговое окно **Block Definition** (Описание блока)

Для создания блока необходимо определить на изображении **базовую** точку и выбрать объекты чертежа, подлежащие включению в блок. Базовой называется точка, используемая для вставки блока.

Способы ввода данных:

- 1) выбрать объекты на экране (**Select Object**);
- 2) выбрать объекты с помощью функций быстрого отбора AutoCAD по свойствам (**Quick Object**).

Возможные варианты выбора: с сохранением примитивов в текущем состоянии; с заменой в существующем чертеже блоком; с удалением из текущего чертежа при записи в базу данных.

Возможен предварительный просмотр процесса создания изображения блока.

При создании автономного блока дополнительно задается имя и расположение файла автономного блока, а также единицы измерения, которые будут использоваться при вставке блока.

Для вставки блоков используется команда **INSERT** (ВСТАВИТЬ), кото-

рой соответствуют кнопка  панели инструментов *Draw* и пункт *Block* падающего меню *Insert* (Вставка).

**Подготовка и выпуск чертежей.** *Идеология создания чертежей в AutoCAD с применением видовых экранов.* Существуют два принципиально различных подхода к оформлению чертежей в AutoCAD.

Пользователи, работавшие со старыми версиями (до 14-й включительно), привыкли чертить конструкцию в масштабе, необходимом для размещения на листе. Затем виды компоновались в пространстве, помещались внутрь штампа требуемого формата и распечатывались. При этом, если в чертеже содержалось несколько листов, они размещались в пространстве произвольно относительно друг друга.

Этот путь интуитивно понятен и прост, но неудобен тем, что в разных чертежных видах применяется разный масштаб и приходится производить пересчет размеров и масштабирование фрагментов.

Начиная с версии 2000 (релиз 15) и выше, в AutoCAD был введен новый механизм оформления чертежей. Этот механизм использует такие понятия, как *пространство модели* (или *модель*), *пространство листа*, *видовой экран*, *ассоциативные размеры*.

Процесс проектирования и оформления КД можно разделить на следующие этапы:

1. Проектирование (вычерчивание) в пространстве модели конструкции в масштабе 1:1. На данном этапе проектируется конструкция и строится необходимое число видов без нанесения на них пояснительной информации и размеров. Чертежные виды могут располагаться в пространстве модели произвольно.

2. Из шаблона создается необходимое число новых листов необходимого формата с штампом и основной надписью.

3. На каждом из листов создаются видовые экраны, представляющие собой чертежные виды. Для каждого видового экрана настраивается масштаб отображения и другие параметры.

4. Видовые экраны размещаются на листе и закрепляются.

5. В пространстве листа на виды наносятся ассоциативные размеры, а также наносятся все остальные элементы оформления.

К достоинствам данного способа оформления стоит отнести:

– удобство проектирования (нет необходимости чертить в масштабе; нет необходимости точно размещать чертежные виды);

– легкость размещения чертежных видов на листе;

– более «аккуратное» представление многостраничных документов (один лист – одна закладка);

– возможность сохранения для каждого листа уникальных настроек вывода на печать.

*Пространство модели и пространство листа.* Пространством модели называется чертежная область, предназначенная для проектирования изделия. Ее размеры во всех направлениях не ограничены. Рекомендуется в пространстве

модели чертить конструкцию в масштабе 1:1 независимо от ее габаритов. При этом все дополнительные (местные) виды чертятся также в масштабе 1:1, причем их взаимное расположение может быть произвольным.

Пространством листа называется чертежная область, предназначенная для вывода изображения на печать. Ее особенностью является то, что 1 мм пространства листа теоретически должен быть равен 1 мм на бумаге печатающего устройства. На практике получается, что почти все малоформатные принтеры имеют поля печати. Поэтому при выводе на печать имеется некоторая погрешность (уменьшение). Обычно она не превышает 5–8 %.

*Видовые экраны.* Видовой экран (ВЭ) представляет собой фрагмент модели, находящийся на листе. При этом для каждого видового экрана отдельно можно настроить масштаб отображения. В сущности, ВЭ является границей чертежного вида на листе. Видовые экраны бывают двух типов: прямоугольные и произвольной формы. Создание и управление ВЭ производится при помощи инструментов, представленных на панели *Viewports*. На этой панели есть также поле, отображающее масштаб для выделенного ВЭ.

Видовые экраны могут накладываться друг на друга полностью или частично. Для того чтобы границы ВЭ не выводились на печать, их следует помещать на непечатаемый слой.

## 2.9 Трехмерное моделирование в системе AutoCAD

Система AutoCAD 2000 / 2002 / 2004 включает в себя достаточно широкий спектр средств трехмерного моделирования. Она позволяет работать как с простейшими, примитивами, так и со сложными поверхностями и твердыми телами. Базовые типы пространственных моделей, используемых в AutoCAD, можно условно разделить на три группы:

- каркасные модели;
- модели поверхностей;
- твердотельные модели.

*Каркасная модель* – это совокупность отрезков и кривых, определяющих ребра фигуры. В каркасном моделировании используются трехмерные отрезки, сплайны и полилинии, которые позволяют в общих чертах определить конфигурацию изделия – построить его каркас. Данный вид работы следует рассматривать главным образом как этап вспомогательных построений для трехмерного проектирования более высокого уровня.

*Поверхностная модель* – это совокупность поверхностей, ограничивающих и определяющих трехмерный объект в пространстве. Моделирование поверхностей применяется для детальной отработки внешнего облика изделия. Создаваемые при этом объекты характеризуются лишь конфигурацией своей поверхности и поэтому непригодны для решения таких задач, как определение инерционно-массовых характеристик изделия или получение необходимых изображений для оформления чертежей. Область применения данного вида мо-

делирования – дизайн, решение задач компоновки сложных изделий.

Набор средств моделирования поверхностей системы AutoCAD 2002 весьма широк и позволяет создавать пространственные объекты практически любой формы. Имеется возможность создавать следующие основные типы поверхностей: линейчатые поверхности, поверхности Кунса, поверхности Безье.

**Твердотельное моделирование** является основным видом трехмерного проектирования изделий машиностроения. Создаваемые в ходе такого моделирования тела воспринимаются системой как некие единые объекты, имеющие определенный объем. Твердотельное моделирование позволяет не только эффективно решать компоновочные задачи, но и определять инерционно-массовые характеристики, а также получать с пространственного объекта необходимые виды, разрезы и сечения для оформления рабочей документации. Твердотельные модели могут подвергаться различным расчетам, в том числе методом конечных элементов.

Средства твердотельного моделирования системы AutoCAD не позволяют осуществлять параметрическое моделирование. Поэтому возможности этой системы в данной области уступают возможностям таких систем, как *Autodesk Mechanical Desktop*, *Inventor* или *Solid Works*. Тем не менее AutoCAD вполне позволяет создавать твердотельные модели деталей и узлов, конфигурация которых представляет собой набор простейших форм. Серьезные сложности возникают лишь при моделировании изделий сложной неправильной формы, например литых деталей.

Помимо средств создания пространственных объектов, блок трехмерного моделирования системы AutoCAD включает в себя средства просмотра объемного изображения, визуализации и средства редактирования трехмерных объектов.

Необходимо открыть следующие панели инструментов перед началом работы: *View*→*Toolbars*→*3D Orbit*, *View*→*Toolbars*→*Shade*, *View*→*Toolbars*→*Solids*, *View*→*Toolbars*→*Solids Editing*.

Элементарными трехмерными объектами системы AutoCAD, предназначенными прежде всего для каркасного моделирования, являются отрезки, сплайны и полилинии. Первые два типа объектов создаются при помощи тех же команд **LINE** и **SPLINE**, которые используются при плоском черчении. Особенностью при их использовании в трехмерном пространстве является то, что при задании координат точек следует указывать и координату оси Z (при плоском черчении эта координата опускается). При указании точек графическим курсором следует пользоваться объектной привязкой, так как в противном случае система воспринимает указанные точки, лежащие на плоскости XY текущей системы координат. Трехмерные полилинии создаются при помощи специальной команды **3DPOLY**.

Некоторые средства построения и редактирования поверхностей указаны в таблицах 2.2, 2.3, а средства построения и редактирования твердотельных объектов – в таблицах 2.4, 2.5.

## Моделирование поверхностей. Средства построения поверхностей

Таблица 2.2 – Команды построения поверхностей

Команда	Назначение команды	Вызов команды
1	2	3
<b>3DFACE</b>	Построение пространственных трех- и четырехугольных граней	<i>Draw → Surfaces → 3D Face</i>
<b>AI_BOX</b>	Построение поверхности прямоугольного параллелепипеда, основания которого параллельны плоскости XY текущей системы координат	<i>Draw → Surfaces → 3D Surfaces → Box3D</i>
<b>AI_WEDGE</b>	Построение поверхности прямой призмы («клина») с основанием в виде прямоугольного треугольника	<i>Draw → Surfaces → 3D Surfaces → Wedge</i>
<b>AI_PYRAMID</b>	Построение поверхности треугольной или четырехугольной пирамиды или обелиска	<i>Draw → Surfaces → 3D Surfaces → Pyramid</i>
<b>AI_CONE</b>	Построение боковой поверхности кругового прямого конуса, в том числе усеченного	<i>Draw → Surfaces → 3D Surfaces → Cone</i>
<b>AI_SPHERE</b>	Построение сферы	<i>Draw → Surfaces → 3D Surfaces → Sphere</i>
<b>AI_DOME</b>	Построение верхней полусферы (поверхности купола)	<i>Draw → Surfaces → 3D Surfaces → Dome</i>
<b>AI_DISH</b>	Построение нижней полусферы (поверхности чаши)	<i>Draw → Surfaces → 3D Surfaces → Dish</i>
<b>AI_TORUS</b>	Построение поверхности тора	Выпадающее меню: <i>Draw → Surfaces → 3D Surfaces → Torus</i>
<b>3DMESH</b>	Построение произвольной поверхности в виде трехмерной, состоящей из четырехугольных сегментов	<i>Draw → Surfaces → 3D Mesh</i>
<b>AI_MESH</b>	Построение четырехугольного участка билинейной поверхности, состоящего из набора элементарных четырехугольных сегментов	<i>Draw → Surfaces → 3D Surface → Mesh</i>
<b>REVSURF</b>	Построение произвольной поверхности вращения	<i>Draw → Surfaces → Revolved Surface</i>
<b>TABSURF</b>	Построение поверхности, получаемой перемещением образующей вдоль направляющего вектора	<i>Draw → Surfaces → Tabulated Surface</i>
<b>RULESURF</b>	Построение линейчатой поверхности, заданной двумя произвольными направляющими	<i>Draw → Surfaces → Ruled Surface</i>
<b>EDGESURF</b>	Построение нелинейчатой поверхности, заданной двумя парами произвольных линий, образующих замкнутый контур	<i>Draw → Surfaces → Edge Surface</i>

## Средства редактирования поверхностей

Таблица 2.3 – Команды редактирования поверхностей

Команда	Назначение команды	Вызов команды
<b>EDGE</b>	Позволяет управлять видимостью отдельных ребер элементарных сегментов поверхностей	<i>Draw → Surfaces → Edge</i>
<b>PEDIT</b>	Позволяет редактировать свойства сложных поверхностей	<i>Modify → Object → Polyline</i>
<b>PROPERTIES</b>	Редактирование свойств поверхностей	<i>Modify → Properties</i>

## Твердотельное моделирование. Средства построения твердотельных объектов

Таблица 2.4 – Команды построения твердотельных объектов

Команда	Назначение команды	Вызов команды
<b>BOX</b>	Построение твердотельного прямоугольного параллелепипеда с ребрами, параллельными осям текущей системы координат	<i>Draw → Solids → Box</i>
<b>SPHERE</b>	Построение твердотельного шара	<i>Draw → Solids → Sphere</i>
<b>CYLINDER</b>	Построение прямого твердотельного цилиндра с осью, параллельной оси <i>Z</i> системы координат	<i>Draw → Solids → Cylinder</i>
<b>CONE</b>	Построение твердотельного конуса	<i>Draw → Solids → Cone</i>
<b>WEDGE</b>	Построение твердотельной прямой призмы («клина») с основанием в виде прямоугольного треугольника, параллельным плоскости <i>XZ</i>	<i>Draw → Solids → Wedge</i>
<b>TORUS</b>	Построение твердотельного тора	<i>Draw → Solids → Torus</i>
<b>EXTRUDE</b>	Построение твердотельного объекта путем "выдавливания" плоского контура	<i>Draw → Solids → Extrude</i>
<b>REVOLVE</b>	Построение твердого тела вращения	<i>Draw → Solids → Revolve</i>

## Средства редактирования твердотельных объектов

Таблица 2.5 – Команды редактирования твердотельных объектов

Команда	Назначение команды	Вызов команды
<b>CHAMFER</b>	Позволяет создавать фаску (скошенную часть кромки) твердого тела	<i>Modify → Chamfer</i>
<b>FILLET</b>	Позволяет скруглять внутренние и внешние углы твердого тела	<i>Modify → Fillet</i>
<b>SLICE</b>	Позволяет разрезать набор тел плоскостью и при необходимости удалить отсеченные части объектов	<i>Draw → Solids → Slice</i>
<b>UNION</b>	Позволяет создать новый твердотельный объект путем объединения нескольких объектов	<i>Modify → Solids Editing → Union</i>
<b>SUBTRACT</b>	Позволяет создать новую область или твердотельный объект путем исключения (вычитания) аналогичных объектов	<i>Modify → Solids Editing → Subtract</i>
<b>INTERSECT</b>	Позволяет создать новый твердотельный объект, который является общей частью всех указанных объектов	<i>Modify → Solids Editing → Intersect</i>
<b>INTERFERE</b>	Позволяет проверить, пересекаются ли тела одного набора с телами другого набора, и в случае наличия пересечения создать новые объекты, являющиеся общей частью обоих наборов	<i>Modify → Solids → Interfere</i>
<b>SOLIDEDIT</b>	Обеспечивает редактирование твердотельных объектов	<i>Modify → Solids Editing</i>
<b>SECTION</b>	Позволяет построить сечения (в виде замкнутых областей) твердотельных объектов	<i>Draw → Solids → Section</i>

## Общие средства редактирования трехмерных объектов

Для редактирования любых трехмерных объектов могут применяться такие универсальные команды, как **PROPERTIES**, **COPY**, **ERASE**, **EXPLODE**, **MOVE**, **OOPS**, **REDO**, **SCALE**, **UNDO**, а также средство редактирования **Grips** (таблица 2.6). Работа со всеми этими средствами ведется так же, как и при **2D** черчении. Использование ряда других команд редактирования имеет ряд особенностей.

Таблица 2.6 – Команды редактирования трехмерных объектов

Команда	Назначение команды	Вызов команды
<b>3DARRAY</b>	Создает в пространстве копии объектов в виде прямоугольного или полярного массива	<i>Modify → 3D Operation → 3D Array</i>
<b>MIRROR3D</b>	Создает зеркальную копию объектов относительно плоскости	<i>Modify → 3D Operation → Mirror 3D</i>
<b>ROTATE3D</b>	Поворот объектов относительно произвольно направленной оси	<i>Modify → 3D Operation → Rotate 3D</i>
<b>ALIGN</b>	Задаёт положение и ориентацию объекта выравнивая его относительно другого объекта	<i>Modify → 3D Operation → Align</i>

При использовании команды **ARRAY** применительно к трехмерным объектам создается массив объектов, все соответствующие точки которых расположены на одной плоскости, параллельной плоскости XY текущей системы координат. Поэтому с помощью данной команды нельзя создать массив, элементы которого имеют различное расположение относительно оси Z.

При использовании команды **MIRROR** применительно к трехмерным объектам создается их зеркальное отражение относительно плоскости, проходящей через заданную ось симметрии и перпендикулярной плоскости XY текущей системы координат.

При использовании команды **ROTATE** применительно к трехмерным объектам производится их поворот относительно оси, проходящей через заданную точку и перпендикулярной плоскости XY текущей системы координат.

При использовании команды **STRETCH** применительно к поверхностям производится изменение положения их узловых точек. Изменение формы твердотельных объектов с помощью данной команды невозможно (действие команды сводится к перемещению объекта).

## 2.10 Пример трехмерного моделирования в AutoCAD

Пример создания *твердотельной модели* крышки пульта управления в AutoCAD (рисунок 2. 34).

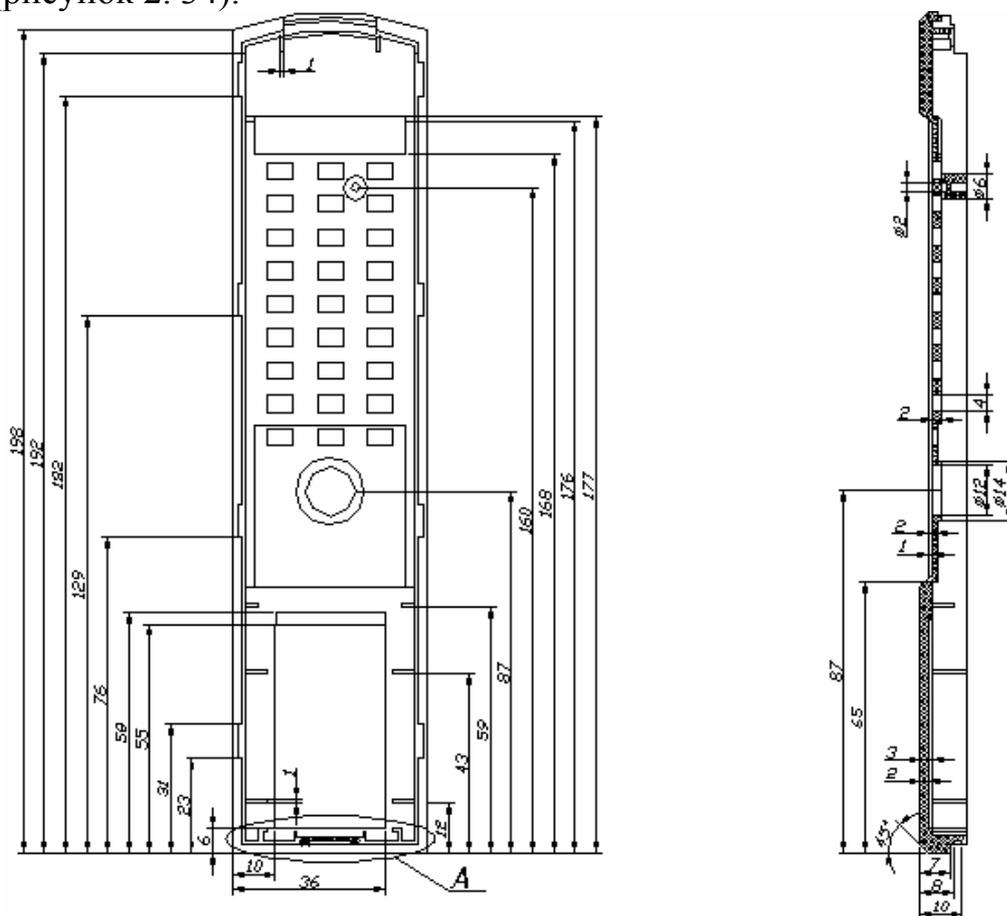


Рисунок 2.34 – Двухмерный чертеж крышки пульта управления

### **2.10.1 Построение внешней формы детали**

**Внимание!** При вводе промежуточных данных для всех команд трехмерного моделирования необходимо его заканчивать нажатием клавиши <Ввод>.

1. Создать и сделать текущим новый слой для построения внешнего контура детали по команде **LAYER**.

2. Построить прямоугольную внешнюю часть пульта.

2.1. По команде **BOX** ввести:

- координаты нижней левой вершины параллелепипеда (0, 0, 0);
- координаты нижней противоположной грани параллелепипеда (198, 48, 0);
- высоту параллелепипеда (8).

2.2. По команде **BOX** ввести:

- координаты (2, 2, 8);
- координаты нижней противоположной грани параллелепипеда (196, 46, 8);
- высоту параллелепипеда (4).

2.3. По команде **UNION** выбрать два ранее созданных объекта.

2.4. По команде **BOX** ввести:

- координаты (66, 0, 0);
- координаты нижней противоположной грани параллелепипеда (172, 48, 0);
- высоту параллелепипеда (2).

2.5. По команде **SUBTRACT** выбрать:

- большую фигуру;
- фигуру, созданную в пункте 2.4.

### **2.10.2 Построение внутренней формы детали**

1. Создать и сделать текущим новый слой для построения внутреннего контура детали по команде **LAYER**.

2. Построить внутреннюю часть пульта.

2.1. По команде **BOX** ввести:

- координаты нижней левой вершины параллелепипеда (4, 4, 4);
- координаты нижней противоположной грани параллелепипеда (194, 44, 4);
- высоту параллелепипеда (8).

2.2. По команде **SUBTRACT** выбрать:

- внешний контур детали;
- фигуру, созданную в пункте 4.2.1.

2.3. По команде **BOX** ввести:

- координаты нижней левой вершины параллелепипеда (4, 4, 2);
- координаты нижней противоположной грани параллелепипеда (64, 44, 2);
- высоту параллелепипеда (2).

2.4. По команде **BOX** ввести:

- координаты нижней левой вершины параллелепипеда (174, 4, 2);
- координаты нижней противоположной грани параллелепипеда (194, 44, 2);
- высоту параллелепипеда (2).

2.5. По команде **SUBTRACT** выбрать:

- внешний контур детали;
- фигуры, созданные в пунктах 2.3 и 2.4.

После промежуточных построений получился вид крышки пульта управления без отверстий под кнопки (рисунок 2.35).

2.6. По команде **BOX** ввести:

- координаты вершины параллелепипеда (98, 9, 2);
- координаты нижней противоположной грани параллелепипеда (102, 13, 2);
- высоту параллелепипеда (2).

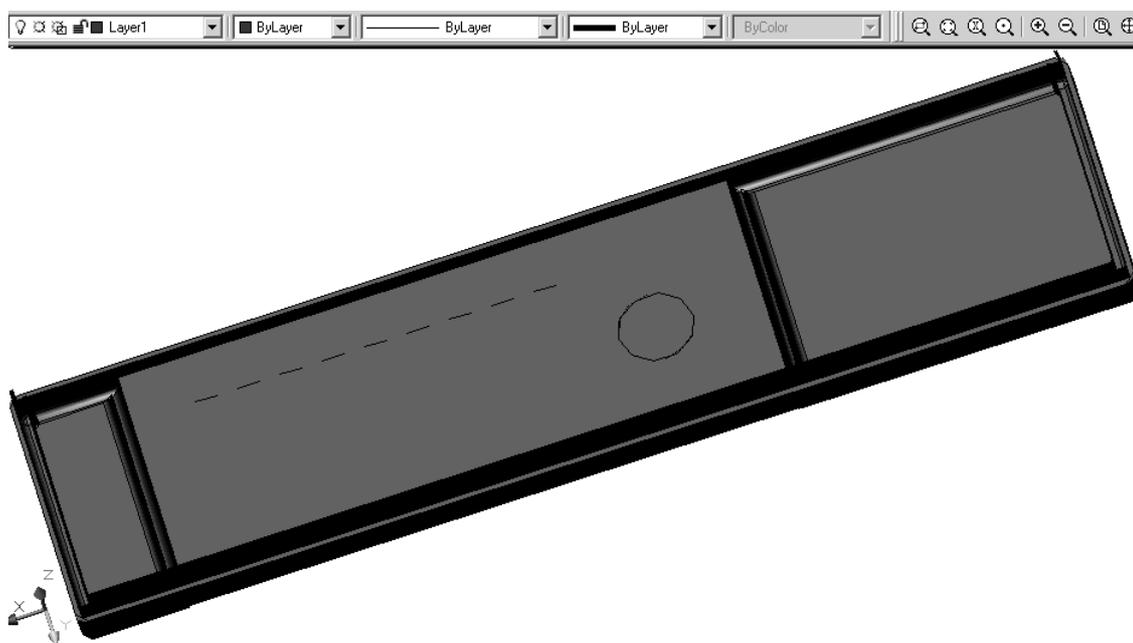


Рисунок 2.35 – Вид крышки пульта управления после промежуточных построений

2.7. По команде **ARRAY** открыть окно и установить указанные на рисунке 2.36 параметры.

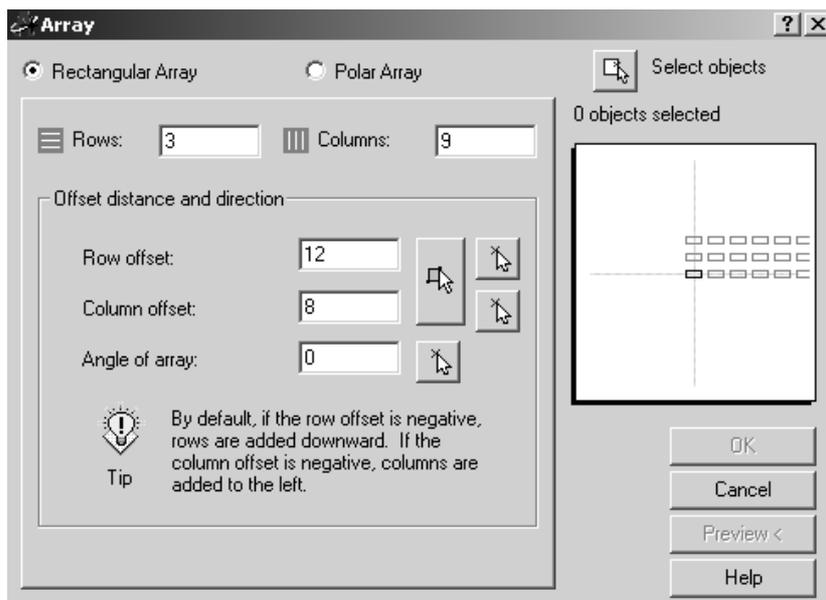


Рисунок 2.36 – Вид окна команды **ARRAY** при моделировании всех отверстий крышки пульта управления

2.8. По команде **SUBTRACT** выбрать:

- внешний контур детали;
- фигуры, созданные в пунктах 2.6 и 2.7.

2.9. По команде **CYLINDER** ввести:

- координаты центра цилиндра (82, 24, 4);
- радиус цилиндра (10);
- высоту цилиндра (2).

2.10. По команде **UNION** выбрать ранее созданные объекты.

2.11. По команде **CYLINDER** ввести:

- координаты центра цилиндра (82, 24, 2);
- радиус цилиндра (7);
- высоту цилиндра (4).

2.12. По команде **SUBTRACT** выбрать:

- контур детали;
- фигуру, созданную в пункте 2.11.

3. Провести скругление углов детали по команде **FILLET**:

- нажать клавиши **<R>** и **<Ввод>**;
- ввести необходимый радиус скругления;
- выбрать угол, который необходимо скруглить.

Трехмерная твердотельная модель крышки пульта управления построена (рисунок 2.37).

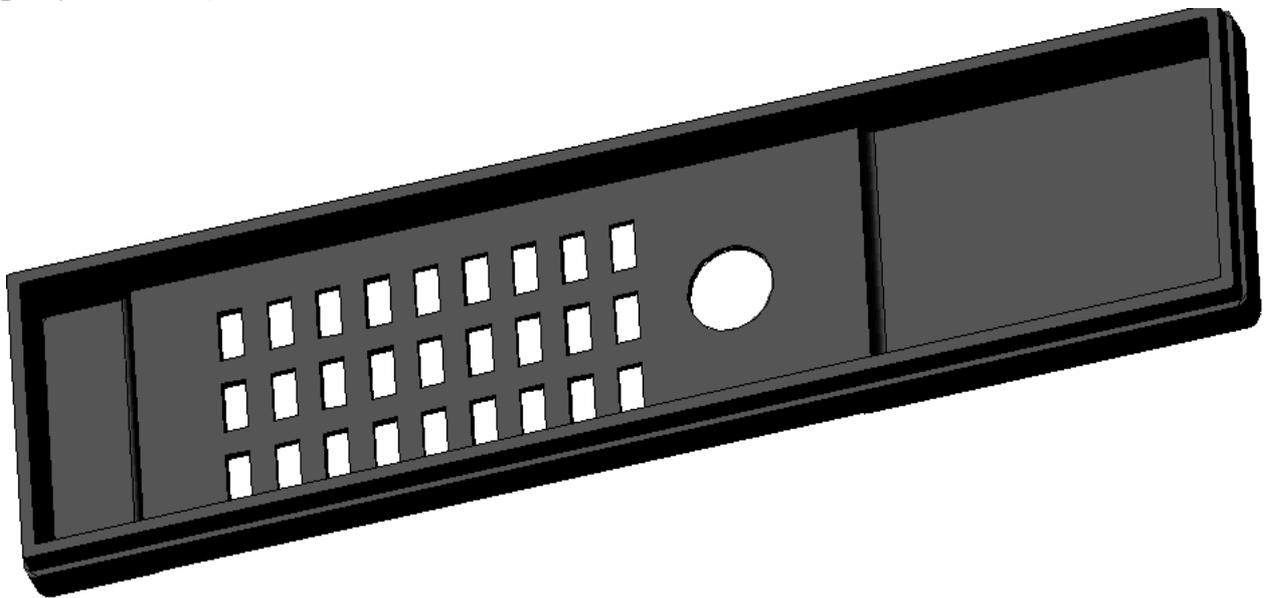


Рисунок 2.37 – Трехмерная твердотельная модель крышки пульта управления со скругленными углами

Аналогичным образом строятся и другие трехмерные твердотельные модели в AutoCAD.

### **3 СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ T-FLEX CAD**

#### **3.1 Общие сведения о программном комплексе T-FLEX CAD**

Система T-FLEX CAD (автоматизированного проектирования и моделирования) успешно применяется в конструировании (проектирование различного оборудования, инструмента; проектирование готовых изделий и т. д.), для решения технологических задач (оформление технологических карт, спецификаций; подготовка данных для разработки технологических процессов; подготовка информации для систем программирования оборудования с ЧПУ), в задачах строительства и архитектуры, при разработке различных типов схем, при динамическом графическом моделировании процессов и механизмов, в задачах художественного оформления и дизайна. Наиболее эффективно T-FLEX CAD применяется в тех областях, где наиболее полно реализуется идея параметрического проектирования, а также, где необходимо охватить все этапы конструирования (эскизный проект, черновой чертеж, рабочий чертеж). T-FLEX CAD позволяет значительно ускорить процесс проектирования и подготовки графической документации.

Параметрическое проектирование, возможность назначения геометрических параметров через переменные и изменение этих параметров - это будущее всех систем автоматизированного проектирования и черчения. Эффективность системы T-FLEX CAD базируется в первую очередь на новой геометрической модели. Эта модель позволяет наполнить понятие "параметризация" более глубоким, чем это принято в других системах, содержанием. Идея параметризации уже прочно завоевала свое место в компьютерном проектировании. Под параметризацией подразумевается, прежде всего, многократное использование чертежа с возможностью изменения его параметров. T-FLEX CAD использует привычные для конструктора элементы и параметры проектирования. При этом совершенно не обязательно ставить элементы чертежа на точные позиции. Возможности по модификации размеров элементов и их положения на чертеже не имеют аналогов в других CAD-системах. Уникальными по своим возможностям являются средства создания сборочных параметрических чертежей. T-FLEX CAD позволяет получать сложные чертежи, в которых его отдельные части могут быть взаимосвязаны. Связь можно задавать как через геометрическую зависимость, так и через значения параметров. При этом обеспечивается удаление невидимых линий в случае, если отдельные части чертежа перекрывают друг друга. Меняя параметры сборочного чертежа, можно за считанные секунды получить готовые чертежи нового проектируемого изделия. Одновременно с измененным сборочным чертежом получают и чертежи его составных частей (деталей), а также другие сопутствующие документы.

Одним из обычных атрибутов параметрических CAD-систем является язык программирования, который используется для задания параметрических связей. В этой связи проявляется еще одно существенное достоинство программы T-FLEX CAD. От разработчика не требуется никаких специальных знаний в об-

ласти программирования. Параметрам чертежа могут назначаться переменные. С помощью простых математических формул можно связывать переменные между собой. При этом не нужно изучать какой-либо язык программирования. Назначение переменных может происходить и при создании элемента, и при его последующем редактировании. Значение переменных можно получать из других чертежей или автоматически выбирать из баз данных. Всё это делает возможности по модификации чертежа безграничными.

T-FLEX CAD предлагает полный набор средств для оформления технических чертежей: нанесение линий различных типов, штриховок, размеров, текстов, шероховатостей, специальных символов и т.д. Важно отметить, что все элементы оформления могут быть связаны с параметрами чертежа. Это означает, что изменение параметров чертежа автоматически приводит к изменению соответствующих элементов оформления. Чертежи могут создаваться в соответствии с требованиями ЕСКД или международных стандартов. T-FLEX CAD позволяет мгновенно перевести готовый чертеж из одного стандарта в другой.

Трёхмерная версия T-FLEX CAD 3D позволяет получать параметрические трёхмерные модели. Созданные в системе трёхмерные твёрдотельные модели легко модифицируются, так как T-FLEX CAD 3D базируется на двумерной версии пакета. При параметрическом изменении двумерного чертежа автоматически изменяется его трёхмерное представление.

Работа с системой понятна даже начинающему пользователю. Каждая команда системы реализована так, чтобы при разработке не возникало никаких затруднений.

Параметрическое проектирование, возможность назначения геометрических параметров через переменные и изменение этих параметров – это будущее всех систем автоматизированного проектирования и черчения. Эффективность системы T-FLEX CAD базируется в первую очередь на новой геометрической модели. Эта модель позволяет наполнить понятие "параметризация" более глубоким, чем это принято в других системах, содержанием.

T-FLEX CAD предлагает полный набор средств для оформления технических чертежей: нанесение линий различных типов, штриховок, размеров, текстов, шероховатостей, специальных символов и т. д. Все элементы оформления могут быть связаны с параметрами чертежа. Это означает, что изменение параметров чертежа автоматически приводит к изменению соответствующих элементов оформления. Чертежи могут создаваться в соответствии с требованиями ЕСКД или международных стандартов. T-FLEX CAD позволяет мгновенно перевести готовый чертеж из одного стандарта в другой. T-FLEX CAD автоматически присваивает созданным документам имя с расширением файла **\*.grb**.

На рисунке 3.1 представлено окно системы T-FLEX CAD с указанием его основных элементов.

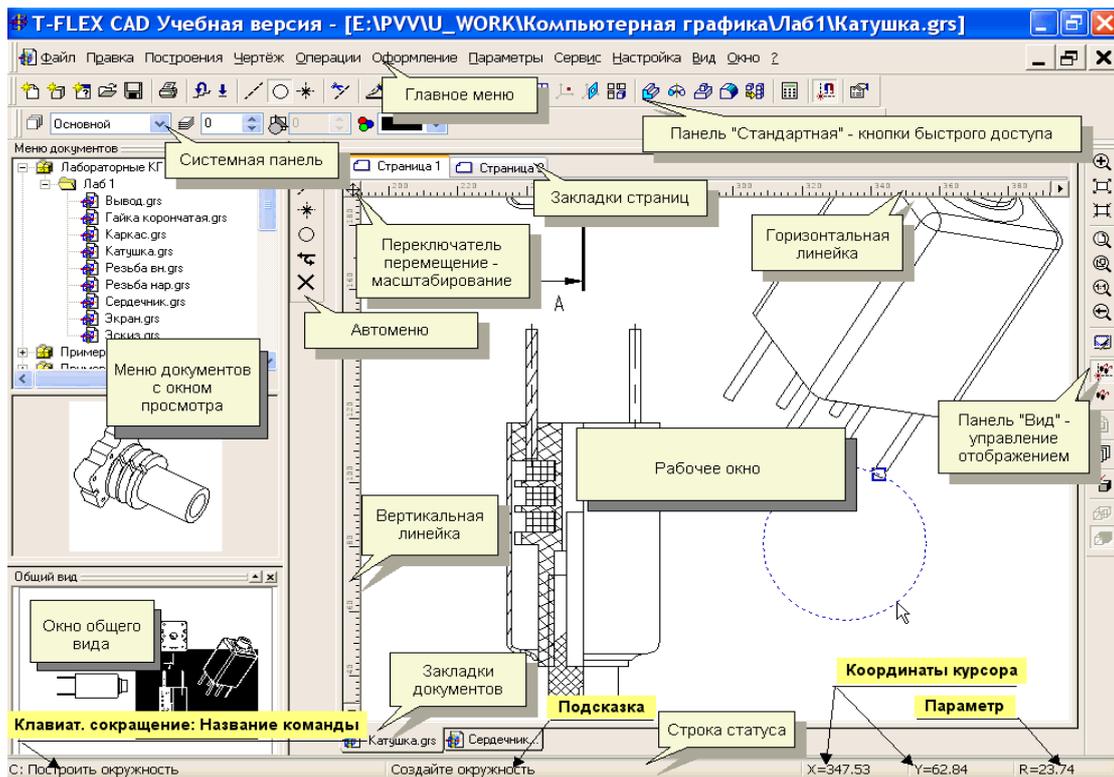


Рисунок 3.1 – Элементы интерфейса системы T-FLEX CAD

Панель «Вид» позволяет выполнить следующие действия по управлению изображением:

- |  |  |
|--|--|
|  | <ul style="list-style-type: none"> <li>Определить размер изображения рамкой;</li> <li>Увеличить изображение;</li> <li>Уменьшить изображение;</li> <li>Вписать в рабочее окно лист чертежа;</li> <li>Вписать в рабочее окно все элементы изображения (элементы построения не учитываются);</li> <li>Отобразить в натуральную величину;</li> <li>Вернуться к предыдущим границам экрана (до десяти шагов);</li> <li>Обновить изображение;</li> <li>Скрыть / показать элементы построения;</li> <li>Спрятать / показать элементы, задаваемые пользователем;</li> <li>Открыть чертежный вид;</li> <li>Работа с листами документа.</li> </ul> |
|--|--|

Линейки рабочего окна можно использовать для сдвига изображения (рисунок 3.2, а) или для его плавного масштабирования (рисунок 3.2, б). Переключение режимов осуществляется щелчком мыши на кнопке, показанной на рисунке стрелкой.

Окно общего вида (рисунок 3.2, г) отображает часть чертежа, расположенную в пределах рамки формата. В окне общего вида инверсная область показы-

вает текущие границы рабочего окна. С помощью мыши здесь можно задать новые границы отображения.

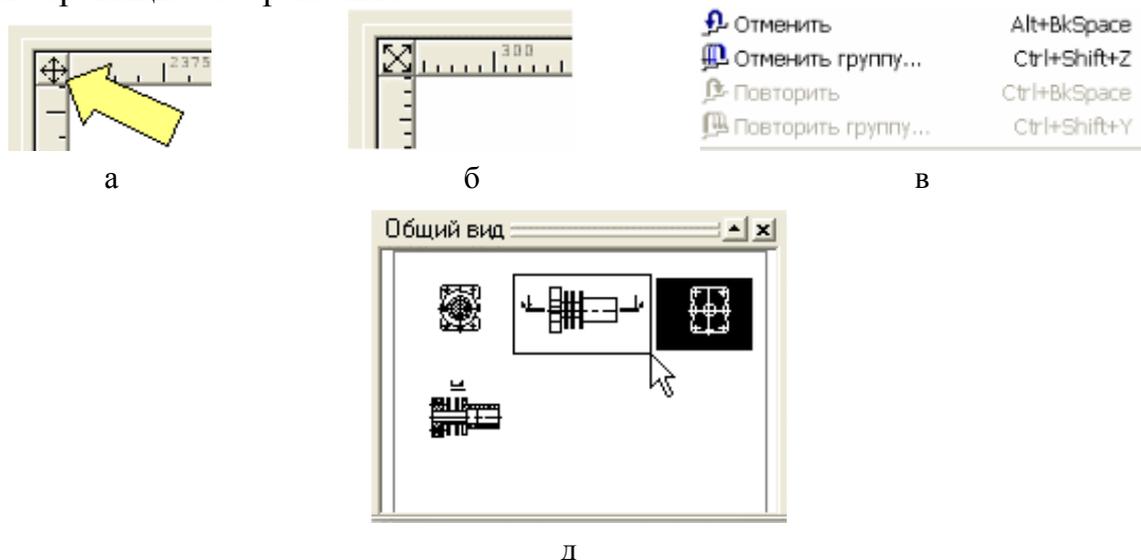


Рисунок 3.2 – Элементы рабочего окна

Показанная группа команд меню **Правка** (рисунок 3.2, в) предназначена для отмены ошибочно выполненных действий или для возврата отмененных.

Команды отката представлены кнопками  в панели «Стандартная». По умолчанию система помнит 50 последних команд. Длину очереди команд можно изменить через меню **Настройка/Установки/Разное**.

Система T-FLEX CAD использует при создании чертежа несколько типов элементов.

**1. Элементы построения** формируют каркас чертежа. С ними связаны элементы изображения, которые и являются тем реальным изображением, которое мы хотим в итоге получить. К элементам построения относятся **линии построения и узлы**. Линии построения и узлы – основные элементы, формирующие параметрическую модель чертежа. По аналогии с черчением их можно сравнить с тонкими карандашными линиями, которые затем обводятся тушью. С помощью задания различных типов линий построения и узлов устанавливается взаимосвязь элементов построения и определяется порядок расчета их положения при параметрическом изменении чертежа. Они присутствуют только на экране и не выводятся на принтер или плоттер.

**Линии построения** – это прямые, окружности, эллипсы, кривые (сплайны), эквидистанты и кривые, заданные функцией. Это базовые элементы параметрической модели в T-FLEX CAD. Они представляют собой тонкие конструкционные линии, с помощью которых вы создаете параметрический каркас вашего чертежа.

**Узлы** – точки пересечения линий построения. Узлы являются базовым элементом создания параметрической модели в T-FLEX CAD. Узлы напрямую участвуют в построении параметрической модели при задании типов линий построения.

**“Свободный” узел** – узел, положение которого определяется абсолютными координатами.

**“Узел с фрагмента”** – узел, который определяется положением фрагмента на чертеже.

**2. Элементы изображения** формируют чертеж. К элементам изображения относятся линии изображения, размеры, тексты, штриховки, допуски формы и расположения поверхностей, надписи, шероховатости. Они могут “привязываться” к элементам построения. В этом случае, при изменении положения линий построения и узлов, элементы изображения изменяют свое положение, что и является основной идеей параметризации в T-FLEX CAD. Эти элементы составляют изображение чертежа при выводе на принтер и плоттер.

**Линии изображения** – отрезки, дуги окружностей, окружности, дуги эллипсов, эллипсы, кривые. Могут быть различных типов (сплошные основные, сплошные тонкие, штриховые, штрихпунктирные и т.д.). Они привязываются к узлам и линиям построения.

**Штриховки, заливки, штриховки по образцу** – замкнутые одноконтурные или многоконтурные области заполненные различными способами. Контур штриховок привязывается к узлам и линиям построения. При изменении положения узлов изменяются контуры штриховок. При этом автоматически изменяется заполнение штриховок в соответствии с изменением контуров.

**Тексты** – однострочная и многострочная текстовая информация, задаваемая в текстовом редакторе и отображаемая на экране различными шрифтами. Положение текстов может быть задано в абсолютных координатах, то есть независимо от элементов построения. Тексты могут быть привязаны к линиям построения и узлам.

**Размеры** – стандартный элемент оформления чертежей. Размер состоит из совокупности линий и текстовой информации. Он может быть построен только при наличии линий построения и узлов. Система T-FLEX CAD поддерживает простановку размеров нескольких стандартов: ЕСКД, ANSI, архитектурный ANSI. Размеры автоматически перестраиваются при параметрическом изменении чертежа.

**Шероховатости** – стандартный элемент оформления чертежей. Состоит из совокупности линий и текстовой информации. Шероховатость может быть привязана к абсолютным координатам, к узлу, к линии построения и к размеру.

**Допуски формы и расположения поверхностей** – стандартный элемент оформления чертежей. Состоит из совокупности линий и текстовой информации. Привязывается только к узлу.

**Надписи** – стандартный элемент оформления чертежей. Состоит из совокупности линий и текстовой информации. Надпись может быть привязана к абсолютным координатам, к узлу, к линии построения.

К сложным элементам изображения относятся фрагменты и картинки.

**Фрагменты** – чертежи системы T-FLEX CAD, которые могут использоваться в других чертежах, для получения составных (сборочных) чертежей. Фрагментом может быть любой чертеж системы T-FLEX CAD. Под параметри-

ческим фрагментом системы T-FLEX CAD понимается чертеж, при нанесении которого на другой чертеж, вы можете задать его положение и параметры, от которых зависит его изображение. При создании параметрических фрагментов необходимо соблюдать определенные правила.

**Картинки** – графические изображения чертежей системы T-FLEX CAD и других систем, сохраненные в различных форматах. Картинки, как и фрагменты, можно использовать в других чертежах. При этом изображение картинок будет неизменным, вы можете только задать его положение и масштаб изображения.

**3. К вспомогательным элементам** относятся **переменные, поля, базы данных, отчеты**, а также некоторые другие служебные данные.

**Переменные** – элементы системы, имеющие имя и значение. Значения переменных можно изменять. Основное назначение переменных – это использование их значений в качестве параметров линий построения. Например, в качестве параметра прямой, параллельной заданной и расположенной от нее на каком-то расстоянии, можно использовать не только число, но и переменную.

**Базы данных** – таблицы, содержащие информацию в упорядоченном виде. Базы данных используются для хранения информации, необходимой для чертежа.

**Отчеты** – текстовые документы, которые создаются с помощью текстового редактора системы T-FLEX CAD и могут включать в себя переменные системы. Служат для создания разнообразных текстовых документов.

### **3.2 Особенности двумерного проектирования конструкций в T-FLEX CAD**

В системе T-FLEX CAD изображение чертежа наносится на основе предварительно созданных элементов построения (см. руководство пользователя). Элементы построения, формирующие каркас чертежа, представляют собой бесконечные прямые и окружности, связанные между собой различными геометрическими отношениями. Этими отношениями могут быть параллельность, касание, симметрия, угол наклона и т.д. По аналогии с черчением на бумаге элементы построения можно сравнить с тонкими карандашными линиями, которые потом обводятся тушью.

Таким образом, прежде чем нарисовать линии чертежа, необходимо с помощью линий построения создать каркас чертежа. Линии изображения чертежа оказываются "привязанными" к линиям построения, и при изменении их положения будут следовать за ним. При этом сохраняются все геометрические отношения, которые были заданы для элементов построения.

В системе T-FLEX CAD чертеж может быть построен одним из следующих способов: параметрически и непараметрически.

## Принципы параметризации чертежа

Параметрический способ построения чертежей – это основной режим работы системы T-FLEX CAD. Используя преимущества параметрического проектирования T-FLEX CAD, возможно создать чертеж, который будет легко изменяться при дальнейшем проектировании. Этот чертеж можно включить в качестве элемента параметрической библиотеки для использования его в других, более сложных, чертежах.

В отличие от эскизного черчения, при создании параметрической модели сначала выполняется разметка чертежа «в тонких линиях». Для этого используются элементы построения: прямые, узлы, окружности и др.

При вводе элемента построения задаются геометрические взаимосвязи, например, параллельность существующему элементу, и числовые параметры (для параллельной прямой – расстояние между элементами). В дальнейшем числовые параметры можно изменять, тем самым, модифицируя модель.

Значение параметра можно задать вводом константы, или связав параметр с переменной. В последнем случае управление моделью становится более удобным, так как доступ ко всем переменным осуществляется из специального редактора. Кроме того, наличие переменных дает возможность вводить в модель различные расчеты.

Ввод значения параметра осуществляется в окне свойств элемента (в данном случае элементом является прямая, параллельная другой прямой, и параметром, соответственно, будет расстояние между ними). Вместо значения в данном окне можно указать имя переменной. Если такой переменной не существует, будет запущен диалог создания переменной. Переменные могут быть предварительно созданы в **Редакторе переменных**. Не все элементы построения имеют числовые параметры. Так, прямая, проходящая через два узла, полностью определена геометрическими связями.

При создании параметрического чертежа очень важно заранее определить последовательность построения и взаимосвязи между отдельными элементами. Здесь может помочь представление о последовательности изготовления детали – выбираются технологические базы, от них задаются общие габариты и отдельные элементы детали, часть размеров задается от полученных элементов и т.д. Аналогично надо действовать и при создании модели. Грамотно проставленные на чертеже детали размеры несут в себе информацию о последовательности ее изготовления.

Так как одну и ту же задачу можно решить множеством способов, искусство инженера-проектировщика электронной аппаратуры заключается в построении наиболее простой и понятной модели.

Построение чертежа в T-FLEX CAD начинается с задания элементов построения. Элементы построения могут быть созданы различными способами. Сначала задаются базовые линии построения, от которых в дальнейшем строятся новые линии построения. Базовыми линиями могут быть вертикальные и горизонтальные прямые. Далее происходит построение прямых или окружно-

стей, зависимых от базовых. Например, параллельные прямые, окружности касательные прямым. Тем самым определяется способ построения новых линий, который запоминается в системе. На пересечении построенных прямых создаются узлы, которые требуются для проведения дальнейших построений. Затем продолжается построение прямых и окружностей, заданием их различными способами относительно построенных ранее. Например: прямая, проходящая через два узла, окружность касательная к прямой и проходящая через узел и т.д. Все эти способы сохраняются, и в дальнейшем при изменении базовых или других элементов построения положение зависимых прямых, окружностей и узлов будет определяться исходя из способа их задания.

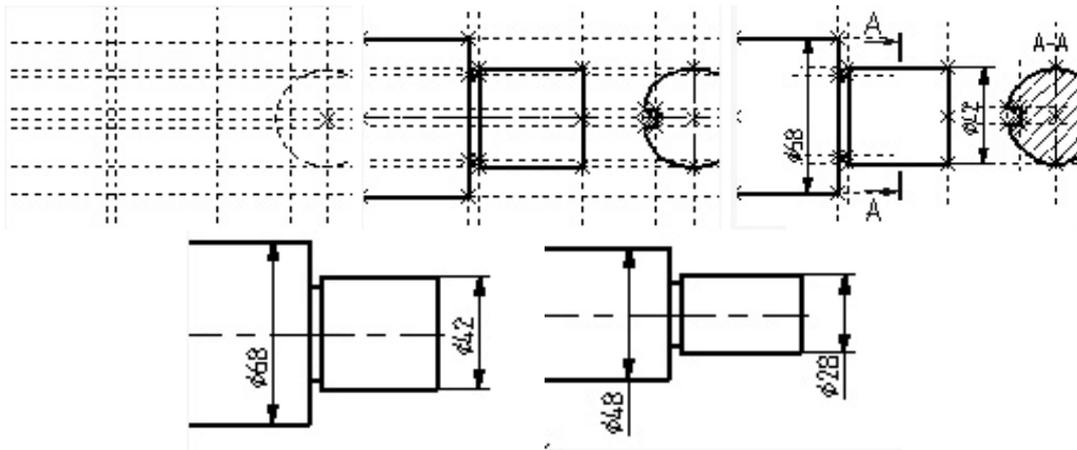


Рисунок 3.3 – Иллюстрация последовательности построения чертежа

Таким образом, на начальном этапе построения чертежа определяются параметрические зависимости элементов построения чертежа, то есть строится параметрический каркас чертежа.

После задания вспомогательных линий осуществляется нанесение элементов, формирующих изображение чертежа. Наносятся линии изображения - отрезки, дуги, окружности. При их нанесении происходит привязывание их к созданным элементам построения - узлам и линиям построения.

После нанесения основного изображения осуществляется переход к оформлению чертежа. Наносятся размеры, которые привязываются к линиям построения и узлам. Определяются контуры штриховок и способы их заполнения. Наносится текстовая информация. При нанесении текстов можно задать привязку текстов к элементам построения - узлам, линиям построения. Это необходимо сделать, если текст должен перемещаться вместе с изменением изображения чертежа. Далее, если это необходимо, наносятся допуски, шероховатости, надписи. После этого получается параметрический чертеж с возможностью его модификации. Существует возможность изменять параметры элементов построения. Например, расстояние параллельной линии от базовой, угол наклона прямой относительно другой, радиусы окружностей.

При этом все элементы нанесения будут изменять свое положение вслед за изменением положения элементов построения, с которыми они связаны. Таким

образом, будут получаться различные варианты одного и того же чертежа. При этом все оформление чертежа будет соответственно изменяться. И все это за очень короткое время.

Линии построения могут иметь числовой параметр (расстояние от параллельной прямой, угол, радиус окружности и т.д.). Значение параметра задается во время создания элемента построения. Некоторые из конструктивных элементов могут не иметь параметра, например, прямая, проходящая через два узла. В качестве параметра элемента построения может использоваться переменная. Значение переменной может быть задано константой или выражением. В выражении могут присутствовать другие переменные, различные арифметические и логические операции, условные операции, обращения к стандартным математическим функциям, а также обращения к внешним базам данных.

Любой чертеж системы T-FLEX CAD может быть использован при работе с другим чертежом в качестве фрагмента, с соответствующими параметрами. Фрагменты используются для создания параметрических сборочных чертежей.

Непараметрический чертеж строится аналогично большинству известных систем, когда сразу наносятся линии изображения.

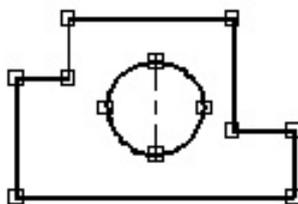


Рисунок 3.4 – Пример непараметрического чертежа

В системе T-FLEX изображение чертежа наносится на предварительно созданные элементы построения. Элементы построения, формирующие каркас чертежа, представляют собой бесконечные прямые, окружности, сплайны, эллипсы, эквидистанты, связанные между собой различными геометрическими отношениями. Этими отношениями могут быть параллельность, касание, симметрия, угол наклона и т.д. По аналогии с черчением на бумаге элементы построения можно сравнить с тонкими линиями, которые затем обводятся тушью. Прежде чем нарисовать линии чертежа, необходимо с помощью линий построения задать каркас чертежа. Линии изображения чертежа затем наносятся на этот каркас. Линии изображения чертежа оказываются "привязанными" к линиям построения, и при изменении их положения будут следовать за ними. При этом сохраняются все геометрические отношения, которые были заданы для элементов построения.

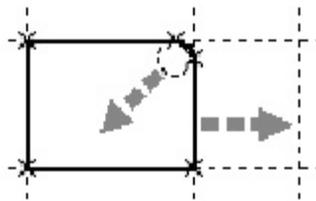


Рисунок 3.5

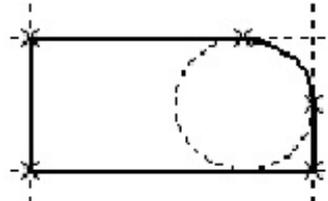


Рисунок 3.6

Как видно на рисунке выше перемещение правой линии построения и увеличение радиуса окружности сопряжения никак не отразились на геометрических связях чертежа.

Такой подход, когда элементы чертежа взаимосвязаны друг с другом, позволяет получить полностью организованный чертеж.

Элементами построения в T-FLEX являются линии построения - прямые, окружности, эллипсы, сплайны, а также узлы. Узлы - это точки пересечения линий построения. Элементы построения не выводятся на принтер или плоттер. Кроме того, в любой момент их можно сделать невидимыми.

Важно отметить, что прежде чем создавать элементы построения вашего чертежа, необходимо проанализировать, какие именно отношения между линиями вы хотите задать. Ведь именно от этого будут зависеть возможности вашего чертежа к параметрическим изменениям.

Параметрам чертежа, а фактически параметрам линий построения (расстояниям, диаметрам и т.д.), можно назначать переменные. Это можно осуществлять и во время создания линии построения, и во время ее последующего редактирования. Затем, задавая значения переменных, можно получать новые варианты чертежа. С помощью простых математических формул в редакторе переменных переменные можно связывать между собой.

При изменении положения какого-либо элемента построения или значения какой-либо переменной система производит пересчет чертежа и его последующую перерисовку. Пересчет производится в соответствии с теми геометрическими отношениями, которые были заложены при его создании, а также, исходя из математических связей между переменными, если они были заданы. Как правило, грамотно созданный параметрический чертеж содержит несколько ключевых (внешних) переменных, в зависимости от которых производится перерасчет остальных переменных и всего чертежа.

Методика и разнообразные примеры построения прямых, окружностей, эллипсов, сплайнов, эквидистант, кривых, путей, узлов, а также возможных вариантов расположения их относительно друг друга приведены в разделе «Руководство пользователя».

Чертежи системы T-FLEX можно объединять друг с другом. Например, можно к чертежу детали добавить чертеж бланка форматки. Или можно включить чертеж болта в какой-нибудь сборочный чертеж. Более того, T-FLEX позволяет менять параметры включаемого чертежа. Так, если при включении болта задать ему необходимый диаметр, то на сборочном чертеже вы получите со-

ответствующее изображение болта, рассчитанное на основе этого диаметра.

Чертежи T-FLEX, которые вы включаете в свой чертеж, называются фрагментами. Как и для других элементов, формирующих чертеж, для фрагментов существует команда создания (то есть включения в текущий чертеж) - FRagment и команда редактирования - EFRagment.

Чертежи, полученные с использованием чертежей-фрагментов, мы будем называть сборочными чертежами. Сборочные чертежи непосредственно не содержат в себе данных фрагментов. Они хранят лишь ссылки на эти чертежи. Это позволяет добиться максимально компактного хранения чертежей в памяти и на диске. Кроме того, если один и тот же чертеж был включен в различные чертежи, то при его модификации произойдут соответствующие изменения и в тех чертежах, в которых он был использован.

Создание чертежей с помощью фрагментов в ряде случаев позволяет добиться ряда преимуществ. Во-первых, для сложных чертежей упрощается процесс создания, так как можно сначала создать отдельные части этого чертежа, а затем объединить их. Во-вторых, если создавать параметрический чертеж на основе деталей, составляющих его, то при изменении каких-либо параметров сборочного чертежа вы сможете одновременно получить и полный набор соответствующих этим параметрам чертежей-деталей. В-третьих, в качестве фрагментов удобно создавать часто повторяющиеся элементы чертежей.

Одним из преимуществ системы T-FLEX перед другими CAD-системами является возможность создания параметрических сборочных чертежей, в которых отдельные части-детали связаны между собой. При модификации чертежа сохраняются все основные зависимости между параметрами отдельных деталей. Более того, одновременно с модифицированным сборочным чертежом вы получите полный набор чертежей-деталей, которые участвовали в сборочном чертеже. При этом они будут иметь те параметры, которые получились в результате пересчета сборочного чертежа.

Чертежи-фрагменты, которые участвуют в сборочном чертеже, могут быть переменными, то есть в зависимости от каких-либо условий в сборочном чертеже будет участвовать тот или иной чертеж-фрагмент.

Каждый фрагмент может в свою очередь также состоять из фрагментов, причем уровень вложенности фрагментов практически не ограничен.

На основе фрагментов, входящих в состав сборочного чертежа, можно создать детализированный чертеж.

### **3.3 Трехмерное твердотельное проектирование**

Система T-FLEX CAD позволяет создавать трехмерные (3 D) модели как на основе уже существующих двумерных чертежей, так и непосредственно в трехмерном пространстве. Для отображения трехмерной модели используется **Окно 3D вида**. Окно 3D вида вместе со всеми трехмерными объектами образует **3D сцену**.

Основными видами представления трехмерных объектов являются: *реберная модель* и *рендеринг*. *Реберная модель* дает слабое представление о форме тела, однако полезна в тех случаях, когда при выполнении операций требуется указание элементов внутри тела. *Рендеринг* дает реалистическое отображение с учетом характеристик материала и углов отражения. При рендеринге не учитывается взаимное влияние тел друг на друга – тени, переотражения, а также свойства среды – рассеяние света, туман и т.п. Для получения фотореалистического изображения, учитывающего указанные эффекты, используется программа «*POV-Ray for Windows*», поставляемая вместе с системой T-FLEX CAD.

Для создания трехмерной модели необходимо наличие одной или нескольких **рабочих плоскостей**, предназначенных для задания в пространстве положения вспомогательных элементов и самой трехмерной модели.

Первоначально трехмерные тела получают перемещением в пространстве плоского или пространственного контура (**3D профиля**) по заданной траектории (*операции*: *выталкивание, вращение, тело по траектории, спираль*). 3D профиль создается на основе его проекций на двухмерном чертеже, или чертится непосредственно на рабочей плоскости после ее активизации. При наличии ранее созданных тел 3D профиль можно построить, используя элементы этих тел.

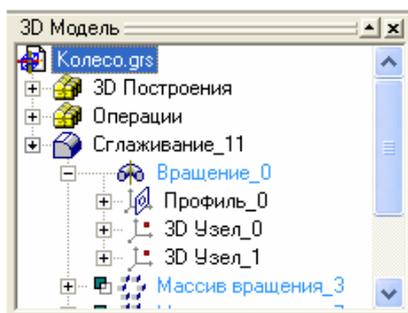
Созданные тела могут быть модифицированы (*операции*: *сглаживание, отсечение, уклон, оболочка*). Над несколькими телами возможно выполнение **булевых операций** (*сложение, вычитание, пересечение*). Кроме того, имеется ряд операций для получения одиночных копий и их массивов.

Система позволяет на основе созданной трехмерной модели автоматически сгенерировать проекционный чертеж с заданными видами. После указания на этом чертеже линий разреза можно получить требуемые сечения и разрезы.

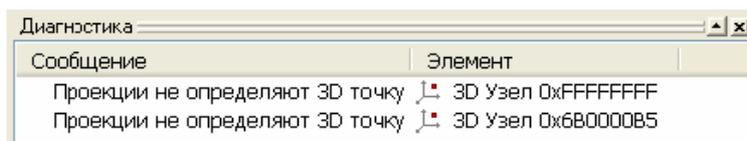
Получаемая трехмерная модель является *параметрической*, поскольку, как двухмерные построения на рабочих плоскостях, так и сами операции создания и модификации тел могут управляться заданием константных значений или переменных. Таким образом, изменение параметров позволяет управлять моделью. Проекционные чертежи, созданные на основе трехмерной модели сохраняют связь с последней, и также будут изменяться.

При работе с трехмерной моделью используются: **окно 3D модели** и **окно диагностики** (рисунок 3.7 а, б). Структура 3D модели, отображаемая в виде дерева в окне **3D модель**, позволяет получить информацию о составе модели и взаимодействии ее элементов. Через дерево модели можно получить доступ к свойствам тех, или иных элементов и операций. Это особенно полезно, если соответствующий элемент недоступен для выбора в окне **3D вида**. Например, после выполнения булевой операции вычитания второе тело пропадает с экрана.

В дереве 3D модели можно выбирать элементы, участвующие в выполнении операций. Это полезно при высокой плотности элементов в 3D окне, когда визуальный выбор затруднен. В окне диагностики выводятся сообщения при некорректных действиях пользователя или при невозможности выполнения действий в силу ограничений системы.



а



б

Рисунок 3.7 – Окно **3D модели** (а) и окно **диагностики** (б)

В силу того, что при выполнении команд создания трехмерных моделей последовательность действий и набор используемых опций выбирает пользователь, выполнение всех этих команд заканчивается нажатием кнопки **ОК** **Автомению**. Следует учитывать, что при вызове новой команды 3D, предыдущая не закрывается. Поэтому перед вызовом очередной команды следует полностью закрыть ранее выполняемую.

Рабочая плоскость является базовым элементом 3D модели. Относительно рабочих плоскостей задаётся положение в пространстве вспомогательных 3D элементов и трехмерных моделей. Схожий с рабочими плоскостями функциональный смысл имеют специальные системы координат, которые в качестве геометрической основы используют не плоскость, а цилиндр, сферу или тор.

Рабочие плоскости - плоскости, ориентированные определенным образом в трехмерном пространстве. Без создания рабочих плоскостей в T-FLEX CAD 3D невозможно построить трехмерную модель. Рабочие плоскости могут быть заданы в 2D окне или в 3D окне. Существуют следующие типы рабочих плоскостей:

- стандартные;
- вспомогательные;
- заданные на основе проекции элементов 3D модели;
- заданные посредством 3D систем координат;
- заданные на основе грани трёхмерного тела;
- заданные на основе другой рабочей плоскости.

Вспомогательные элементы (3D узлы, 3D профили, 3D пути и т.д.) служат для задания формы будущего тела, ориентации в пространстве, определения направлений, векторов, осей, траекторий и т.д.

Твердое тело образуется при выполнении одной из команд "Операции|...". На начальном этапе проектирования это может быть перемещение замкнутого контура по заданной траектории. Перемещение задаётся вектором выталкивания, осью вращения, пространственной траекторией (тело по траектории). Перемещаемый контур может быть плоским или иметь форму трёхмерной поверхности.

Исходными инструментами для создания твёрдых тел является набор базовых операций. Базовые операции использует вспомогательные элементы (3D

профили, 3D узлы, 3D пути и т.д.). В дальнейшем изложении фраза 'выбор операции' подразумевает указание твёрдого тела, созданного с помощью этой операции. Твёрдое тело обладает такими атрибутами как масса, объём, площадь поверхности и т.д..

Работая с 3D моделью пользователь может использовать разные методы проектирования, либо их комбинацию. Создание 3D модели в окне 3D вида описано в разделе "Создание 3D модели в трёхмерном пространстве", проектирование на основе двумерного чертежа - в разделе "Создание 3D модели по 2D элементам". Для работы первым методом необходимо ввести понятие режима активной рабочей плоскости.

Режим активной рабочей плоскости позволяет пользователю работать в 3D окне на рабочей плоскости с любыми 2D командами, создавая элементы построения и изображения так же как при работе с двумерным чертежом.

Элементами 3D модели являются рёбра, грани, вершины тела. Операции над твердыми телами используют элементы 3D модели для создания более сложных тел на основе уже существующих. Таким образом формируется иерархическая структура модели, отражающая исходные 3D элементы и тела и, созданные на их основе.

Создание сборочных конструкций предполагает использование 3D фрагментов или создание в документе нескольких твёрдых тел. В системе T-FLEX CAD любой документ, содержащий 3D модели может являться как сборкой, так и 3D фрагментом (смотри раздел "Создание сборочных трёхмерных моделей"). Гибкий механизм определения привязки элементов 3D сборки позволяет точно указать их положение в сборочной конструкции. Метод, в котором созданные отдельно элементы включаются в сборку, в дальнейшем будем называть "проектированием снизу-вверх". Другой путь заключается в создании 3D модели, содержащей набор операций, и последующим их (операций) сохранением в отдельных документах. Причем пользователь может управлять наличием и типом связи полученных деталей с исходным документов. Такой способ называется "проектированием сверху-вниз".

В некоторых случаях бывает полезно просмотреть сборочную конструкцию в разобранном состоянии. Т.е. когда элементы сборки отделены друг от друга. Для этого у каждого элемента сборки предусмотрены параметры, задающие преобразования (смещение, повороты) относительно системы координат, к которой он привязан. При выполнении команды "Разборка - 3VX " сборочная конструкция будет отображена с учетом заданных параметров разборки.

В системе параметрического твердотельного проектирования T-FLEX CAD 3D существует два оригинальных способа создания трехмерной модели. Первый способ основан на использовании уже имеющегося 2D чертежа конструкции. Для того, чтобы начать создавать 3D модель, необходимо иметь 2D чертеж детали, содержащей необходимое количество видов. Виды должны находиться в проекционной связи между собою. Если вы хотите получить параметрическую трехмерную модель, то нужно задавать параметрические свойства в двумерном чертеже. Первым шагом по созданию модели является создание рабо-

чих плоскостей. Зачастую бывает достаточно создать три взаимно перпендикулярные рабочие плоскости (фронтальную, горизонтальную и профильную), соответствующие виду спереди, сверху и слева. Затем создаются необходимые 3D элементы построений: 3D узлы и 3D профили. 3D узлы используются для привязки 3D профилей, задания векторов выталкивания и осей вращения. Для создания 3D профиля необходимо предварительно создать на 2D виде соответствующие контуры, используя команду "Создать штриховку". Если предполагается использовать штриховку только для 3D построений, то рекомендуется делать ее невидимой, чтобы не загромождать чертеж. Создается 3D профиль следующим образом: выбирается штриховка на 2D виде, указывается какой рабочей плоскости она принадлежит, и с помощью 3D узла задается высота ее расположения над рабочей плоскостью. После этого можно переходить к выполнению операций. Для выталкивания задается контур и два узла, определяющие высоту и направление выталкивания. Для вращения выбирается контур и два узла, определяющие ось вращения. Булева операция позволяет объединить созданные тела в единое целое. Описанный метод - это, можно сказать, классический способ создания параметрических трехмерных моделей в T-FLEX. Но есть и другие методы.

Второй способ основан только на использовании окна 3D вида. Построение модели начинается с создания рабочих плоскостей с последующим созданием на них необходимых 2D элементов. Затем выполняются все необходимые трехмерные операции, после чего можно создать проекции необходимых видов, которые можно дооформить в 2D окне, проставить размеры и т.д. (см. Руководство пользователя).

### **3.4 Создание трехмерной сборки электронного блока с использованием пакета T-FLEX CAD**

В пакете T-FLEX CAD возможно создание двух- и трехмерных сборок. Одним из главных преимуществ T-FLEX CAD перед другими CAD-системами является возможность параметризации элементов, составляющих сборочные чертежи, а также использования ее при создании сборочных чертежей конструкций и модификации проектировочных заданий, когда необходимо часто изменять конструкторские размеры, заменять стандартные изделия и т.д.

При объединении нескольких чертежей друг с другом применяют наложение или совмещение элементов одного чертежа или элемента модели с элементами другого. Такие элементы называются *фрагментами*. Создание чертежей с помощью фрагментов в ряде случаев позволяет добиться ряда преимуществ. *Во-первых*, для сложных чертежей упрощается процесс создания, так как можно сначала создать отдельные части этого чертежа, а затем объединить их. *Во-вторых*, если создавать параметрический чертеж на основе деталей, составляющих его, то при изменении каких-либо параметров сборочного чертежа вы сможете одновременно получить и полный набор соответствующих этим пара-

метрам чертежей-деталей. В-третьих, в качестве фрагментов удобно создавать часто повторяющиеся элементы чертежей.

Основной принцип формирования чертежей из фрагментов заключается в том, чтобы использовать единожды построенный параметрический чертеж и вставлять его в сборочный чертеж, погасив при этом вспомогательные элементы (размеры, некоторые штриховки и т.п.). Сборочные чертежи на основе фрагментов легко преобразовываются при модификации одного из фрагментов, при чем зависимые элементы также преобразовываются. При этом можно получить полный набор чертежей-деталей, которые участвовали в сборочном чертеже. Они будут иметь те параметры, которые получились в результате пересчета сборочного чертежа.

Фрагменты, участвующие в сборке, могут быть переменными, то есть в зависимости от каких-либо условий в сборочном чертеже будет участвовать тот или иной фрагмент. Каждый фрагмент может в свою очередь также состоять из отдельных фрагментов (резьбы, отверстия, стандартные элементы), причем уровень вложенности фрагментов практически не ограничен.

Линии и другие элементы фрагмента не являются частью чертежа, в который он включен. Это означает, что на их основе нельзя обычным способом создавать другие элементы, например, размеры. Для того чтобы решить эту проблему, в системе был создан специальный тип узлов и линий построения. Положение узла с фрагмента определяется узлом чертежа, на основе которого он был создан. Положение линии построения определяется линией изображения фрагмента. Узлы и линии построения, созданные на основе фрагмента могут быть использованы для необходимых операций, как и любой другой элемент чертежа.

Под параметрическим фрагментом системы T-FLEX CAD понимается чертеж, при нанесении которого на другой чертеж, можно задать его положение и параметры, от которых зависит его изображение.

Вставка фрагментов подразумевает использование системы координат фрагмента, называемой *локальной системой координат (ЛСК)*. Локальные системы координат можно создать с помощью *векторов привязки* или *точек привязки*.

Вектор привязки определяет способ привязки чертежа и ориентацию чертежа при вставке его на другой чертеж в качестве фрагмента. При этом вектором привязки задается положительное направление оси X локальной системы координат чертежа. Векторов привязки может быть не более 20.

Векторов привязки создается по команде **Построения / Вектор привязки**



. После вызова команды необходимо указать направление вектора привязки, отметив мышью необходимые точки, задающие направление вектора привязки при вставке в сборку (в сборке ориентацию фрагмента можно изменить). Если для вектора привязки нужно совпадение с осью X, то после указания начала вектора нажмите «**Enter**» или пиктограмму  в **автоменю**. После указания начала и конца вектора привязки появляется меню параметров вектора привязки (рисунок 3.8).

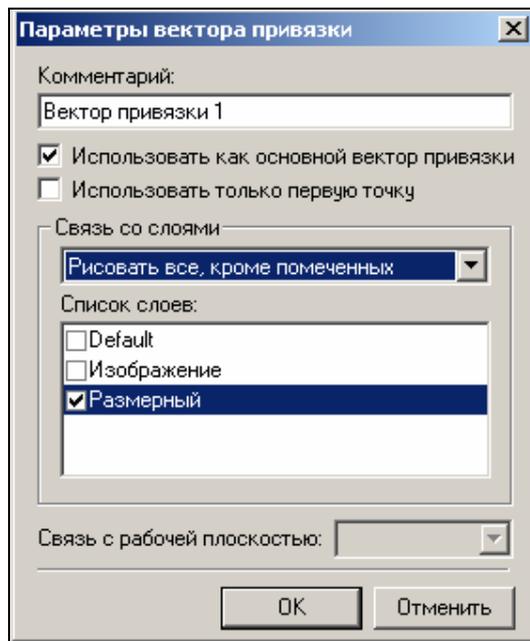


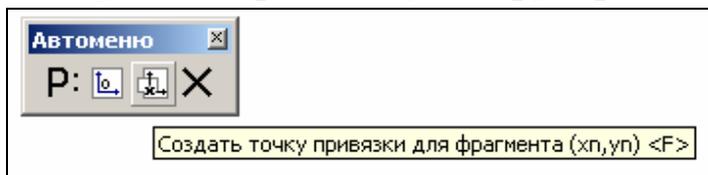
Рисунок 3.8 – Параметры вектора привязки

Как правило, в сборке размеры не требуются, поэтому их «погашают». Одним из способов «погашения» размеров является использование векторов привязки. На рис. 3.1 в поле «Список слоев» выделен слой «Размерный», а в поле связь со слоями указан параметр «Рисовать все, кроме помеченных». Т.о. этот слой не будет отображён на сборке. Существуют и другие способы «погашения» элементов чертежа: с помощью уровней видимости и слоев.

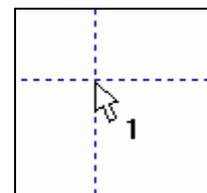
Точки привязки задаются переменными. При этом вертикальным и горизонтальным линиям в качестве параметров задаются зарезервированные имена переменных. В дальнейшем система определяет точку пересечения таких линий как точку привязки фрагмента. Точек привязки может быть несколько.

Фрагменты могут быть поворачиваемыми при вставке в сборку или неповорачиваемыми, а также привязываться одной точкой или несколькими.

Построить точку привязки можно используя специальную опцию . Она становится активной после вызова команды **Построить прямую** (рисунок 3.9, а). При этом переменные создаются автоматически (они по умолчанию являются внешними переменными), а на пересечении линий появляется цифра, соответствующая порядковому номеру переменной (рисунок 3.9, б).



а



б

Рисунок 3.9 – Создание точек привязки

В зависимости от назначения фрагмента можно создать от 0 до 9 точек привязки. Обычно у фрагмента бывают одна точка привязки или две. Первая точка, как правило, задает положение фрагмента на чертеже, вторая – ориента-

цию или размер, либо и то, и другое.

Создание сборки заключается в последовательном нанесении фрагментов. Для того чтобы сборочный чертеж можно было параметрически изменять, необходимо «привязывать» фрагменты к узлам и задавать связи между переменными.

Система T-FLEX CAD позволяет создавать анимацию чертежей, посредством пошагового изменения какого-либо параметра, заданного переменной. Одновременно происходит перерисовка чертежа на каждом шаге. Анимация чертежей выполняется по команде **Параметры / Анимация** . При входе в команду на экране появляется диалоговое окно параметров команды (рисунок 3.10).

При установке параметра **Создать мультимедиа файл** можно создавать файлы формата (\*.avi), здесь же задается «Имя файла» и «Количество кадров в секунду». В данном диалоговом окне возможно определение параметров сжатия созданного файла.

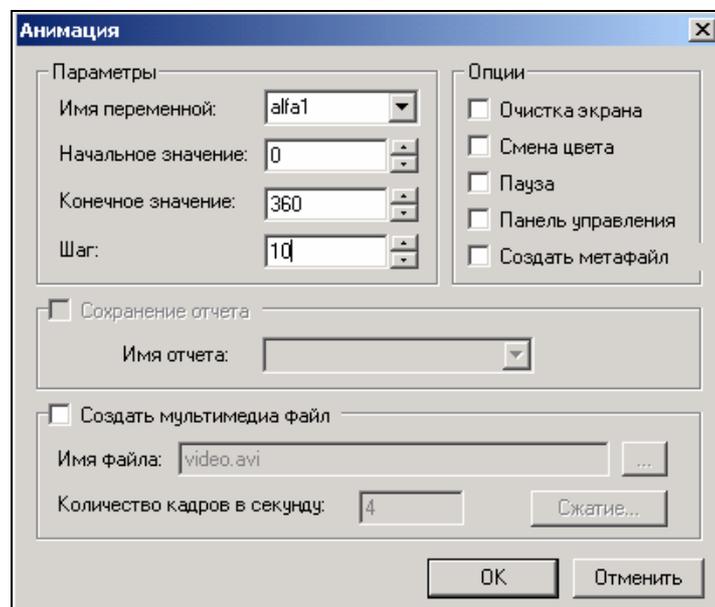


Рисунок 3.10 – Параметры анимации

Аналогично двумерному проектированию, где из плоских чертежей деталей создаются сборочные чертежи, при трехмерном моделировании в качестве фрагмента для создания сборки можно использовать трехмерные модели деталей. 3D фрагменты находятся в прямой связи с двумерным фрагментом, данные которого были использованы для создания 3D фрагмента, т.е. при вставке 3D фрагмента в чертеж добавляется 2D фрагмент, соответствующий ему. Оба фрагмента используют общие данные значений переменных, а также некоторые установки – имя, флаг использования в спецификации, флаг статуса. На основе одного 2D фрагмента можно создать только один 3D фрагмент. Кроме того, нельзя удалить 2D фрагмент, если на его основе был создан 3D фрагмент.

3D фрагмент может быть вставлен на текущий чертеж с помощью команды **3D / Создать / 3D Фрагмент** . Редактируют 3D фрагмент по команде **3D / Изменить операции / 3D фрагмент** .

Для привязки 3D объектов к пространству и друг к другу используются **локальные системы координат**. На вставляемом в сборку фрагменте должна быть **исходная ЛСК**, а в сборочной модели – **целевая ЛСК**. При вставке фрагмента начало координат и оси исходной ЛСК совмещаются с соответствующими элементами целевой ЛСК. Если в модели имеется несколько исходных и целевых ЛСК, то необходимо выбрать одну исходную и одну целевую. Кроме того, в трехмерной модели всегда присутствует мировая система координат, имеющая начало в точке пересечения рабочих плоскостей. Она будет использована для привязки при отсутствии в модели ЛСК (аналогом в двухмерном проектировании является вставка фрагмента без элементов привязки).

При помещении фрагмента на сборку можно задать сдвиги и повороты исходной системы координат относительно целевой по отношению к состоянию их полного совпадения. Для этого существует закладка **Преобразования** в параметрах фрагмента.

Имеется три основных способа построения локальных систем координат:

- **по узлам**, где с помощью 3D узлов задается положение начала координат и, при необходимости, направление осей;
- **по грани**, где начало координат и оси  $X$  и  $Y$  располагаются на указанной грани;
- **по оси цилиндра**, где этой осью задается расположение и направление оси  $Z$  системы координат.

Реально имеется очень большое количество опций для определения и дальнейшего преобразования положения начала координат и направления осей.

ЛСК можно построить заранее, при создании модели детали, а можно строить непосредственно при вставке фрагмента. Для этого в **Автоменю** вставки фрагмента имеется набор соответствующих опций.

Более подробно пакет T-FLEX CAD можно изучить с помощью «Руководства пользователя», а также с помощью специальных обучающих программ, представленных на диске в папке «**Demo\_TFLEX\_CAD**».

## 4 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОГО ТЕХНОЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ

### 4.1 Комплексная автоматизация технологического проектирования

Необходимость комплексной автоматизации – это сокращение сроков:

1. проектирования изделия;
2. технологической подготовки производства (ТПП) его изготовления;
3. настройки технологического оборудования (рисунок 4.1).

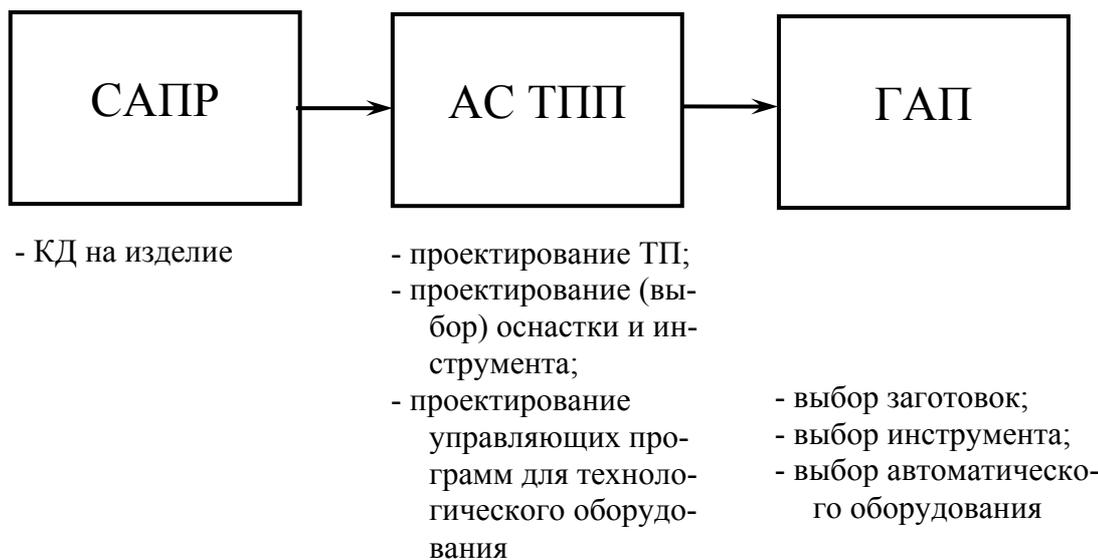


Рисунок 4.1 – Структура комплексной автоматизации ТПП

**Технологическая подготовка производства (ТПП)** — это совокупность организационно–технических мероприятий и инженерно-технических работ, обеспечивающих технологическую готовность предприятия к выпуску изделий заданного уровня качества при установленных сроках, объемах выпуска и затратах.

Значительный вклад в решение проблемы сокращения сроков подготовки производства внесла **Единая система технологической подготовки производства**, которая обобщила и регламентировала (в стандартах) с научных и методических позиций передовые инженерные решения в области ТПП. Достижению этой же цели способствует применение экономико-математических методов и средств вычислительной техники.

**ЕСТПП** - установленная государственными стандартами система организации и управления процессом ТПП, предусматривающая широкое применение прогрессивных унифицированных ТП, стандартной технологической оснастки и оборудования, средств механизации и автоматизации производственных процессов, инженерно-технических и управленческих работ.

*Система предусматривает решение следующих основных задач:*

- обеспечение технологичности конструкции изделия;
- разработка технологических процессов;
- конструирование средств технологического оснащения;
- отладка и внедрение ТП и средств технологического оснащения;
- организация линии, участков и цехов основного производства;
- организация ТПП на базе типовой схемы и типовых положений, а также ее совершенствование;
- управление процессом ТПП во взаимосвязи с другими функциональными подсистемами автоматизированной системы управления производством (АСУП).

**Автоматизированная система ТПП (АСТПП)** — это человеко-машинная система ТПП, в основу которой положен комплекс экономико-математических методов и моделей, организационных форм, электронно-вычислительной и организационной техники с соответствующим оборудованием и программно-математическим обеспечением.

Разработка ТП осуществляется на основе ЕСТПП, которая в свою очередь включает комплекс работ, связанных с технической готовностью предприятия к выпуску нового изделия заданной программы и номенклатуры. Исходные данные для разработки ТП:

- технические условия (ТУ) на изделие;
- комплект конструкторской документации (КД) на изделие;
- программа выпуска изделия (N, шт.);
- плановые сроки освоения производства (Тпл);
- отраслевые, внутризаводские типовые ТП, руководящие технические материалы (РТМ);
- технологические классификаторы деталей, каталоги оборудования и оснастки;
- материальные и трудовые нормативы.

## **4.2 Методы автоматизации технологического проектирования**

Можно выделить следующие методы автоматизированного проектирования ТП (рисунок 4.2):

- 1) на основе типизации и групповой технологии;
- 2) преобразование процессов аналогов;
- 3) многоуровневый итерационный метод.

**Метод 1.** Программы, реализующие метод типизации, характеризуются меньшим на 30 %-40 % объёмом и затратами машинного времени по сравнению с многоуровневым интеграционным методом. Эти программы используются для АП ТП деталей типовых форм (*для остальных деталей проектирование ведётся вторым и третьим методами*). Проектирование сводится преимущественно к параметрической настройке отдельных типовых операций. Применяет-

ся, например, при изготовлении деталей типовых форм, отличающихся только размерами.

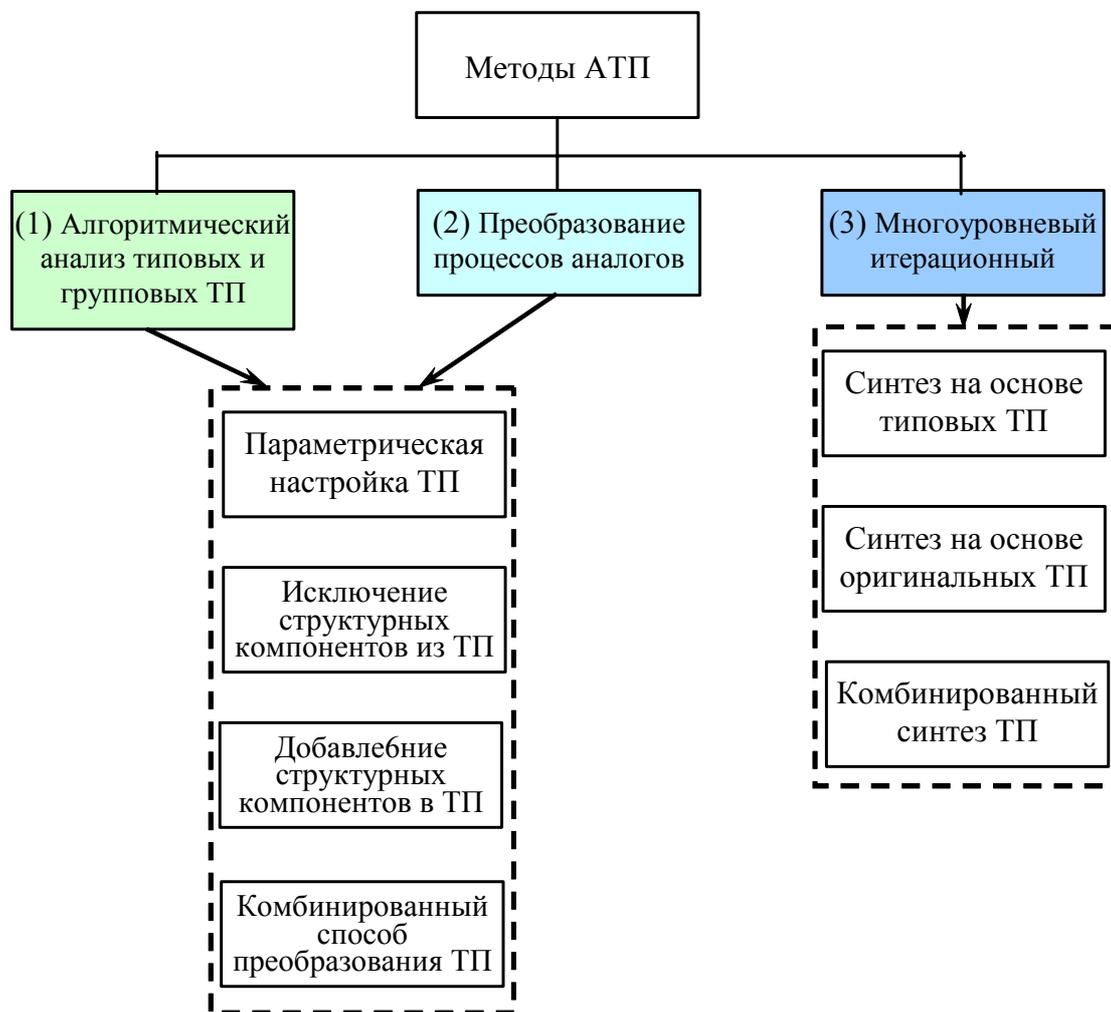


Рисунок 4.2 – Методы автоматизации ТП

**Метод 2.** Использование типового ТП, как процесса-аналога возникает при существенном отличии формы и большего разнообразия обрабатываемых поверхностей. При этом проектирование использует в первую очередь структурное преобразование типовых и групповых ТП.

**Метод 3.** Наиболее общим является многоуровневый интеграционный метод, теоретически разработанный В.Д. Цветковым. Сущность метода основана на принципах (утверждениях):

1. Проектирование ТП можно разделить на четыре последовательно выполняемых уровня (рисунок 4.3).

Степень детализации возрастает от 1-го к 4-му уровню.

2. Процесс проектирования развивается сверху вниз – от синтеза общих принципиальных моделей на первом уровне к проектным решения требуемой степени детализации на следующих уровнях.

3. На всех уровнях, кроме последнего, ввиду недостаточной детализации проектных решений, критерии отбора вариантов носят *эвристический характер*.



Рисунок 4.3 – Уровни проектирования ТП

4. Проектирование на каждом уровне разделяется на совокупность проектных интеграционно взаимосвязанных операций осуществляющих:

- 1) поиск решений аналогов (**П**);
- 2) преобразование процессов аналогов (**Н**);
- 3) синтез различных вариантов (**С**);
- 4) имитационно моделирование (**М**);
- 5) анализ модели (**А**);
- 6) оценка результатов моделирования (**Е**);
- 7) оптимизация (**Q**);
- 8) отбор лучших вариантов (**W**);

Алгоритм проектирования можно представить следующей операторной схемой (рисунок 4.4):

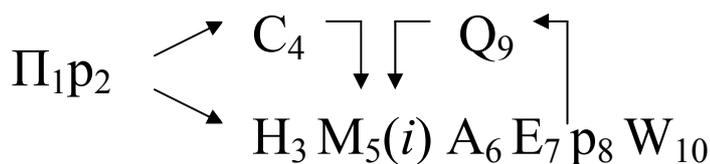


Рисунок 4.4 – Алгоритм проектирования ТП по итерационному методу

Процесс проектирования начинается с операции **поиска П<sub>1</sub>** и нескольких деталей аналогов и ТП аналогов в банке данных.

Если аналогов нет, то *логическим блоком*  $p_2$  управление передаётся на **синтез  $S_4$**  (синтез нового ТП), в противном случае - на **преобразование процессов аналогов  $H_3$** .

Операция **имитационного моделирования  $M_5$**  позволяет прогнозировать характер обработки детали по  $i$ -му варианту, определять *возникающие погрешности и технико-экономические параметры*.

С помощью **операции анализа  $A_6$**  устанавливаются причины отклонений, а **операция оценки  $E_7$**  проверяет выполнение заданных технических требований.

*Логический блок  $p_8$*  осуществляет итерационное построение цикла операций, начиная с *пятой операции (имитационное моделирование)* и включая  **$Q_9$  - оптимизацию**. Этим подготавливается множество вариантов для выбора одного из ТП с помощью **оператора  $W_{10}$**  (Выбор лучшего варианта ТП).

## 5 ТИПОВЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ ДИАЛОГОВОГО ПРОЕКТИРОВАНИЯ ТП

При проектировании технологических процессов применяются:

- 1) *функциональные математические модели;*
- 2) *структурно-логические модели.*

### 5.1 Функциональные математические модели

Для современных технических систем (ТС) характерно:

- наличие значительного числа разнообразных факторов, влияющих на ТС;
- большое число внутренних связей между параметрами ТС и их сложное взаимное влияние;
- наличие нескольких конкурирующих направлений процессов, имеющих различные выходные данные;
- воздействие на ТС большого числа внешних неконтролируемых и неуправляемых факторов, играющих роль возмущений.

Проектирование и оптимизация структуры таких сложных ТС невозможны без применения современных методик моделирования процессов на ЭВМ.

Совокупность физико-химических процессов внутри ТС можно рассматривать как преобразование вводимых потоков энергии и вещества (рисунок 5.1).

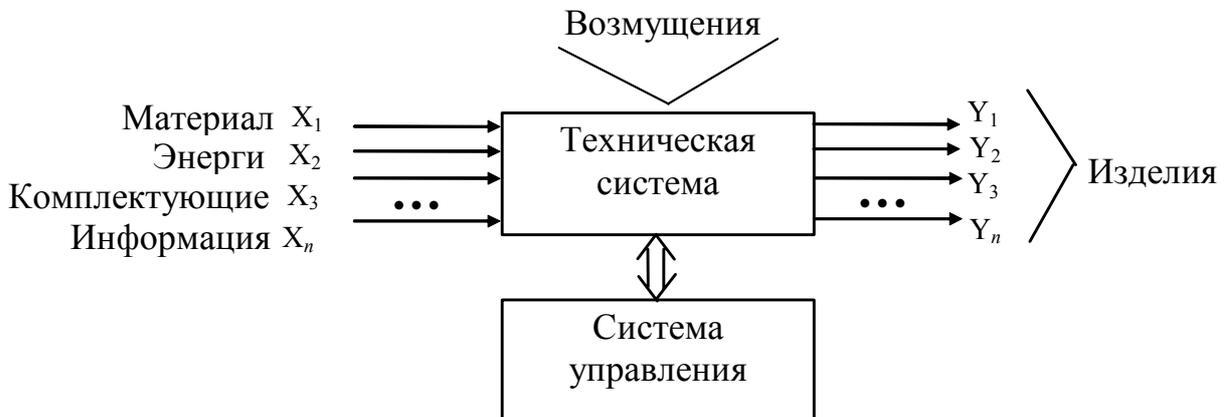


Рисунок 5.1 – Схема сложной технической системы

Для обеспечения требуемых физико-химических превращений в ходе проектирования ТС этими потоками необходимо управлять с помощью *математического моделирования*.

Выраженные через параметры оптимизации целевая функция и ограничения составляют **математическую модель** рассматриваемой задачи.

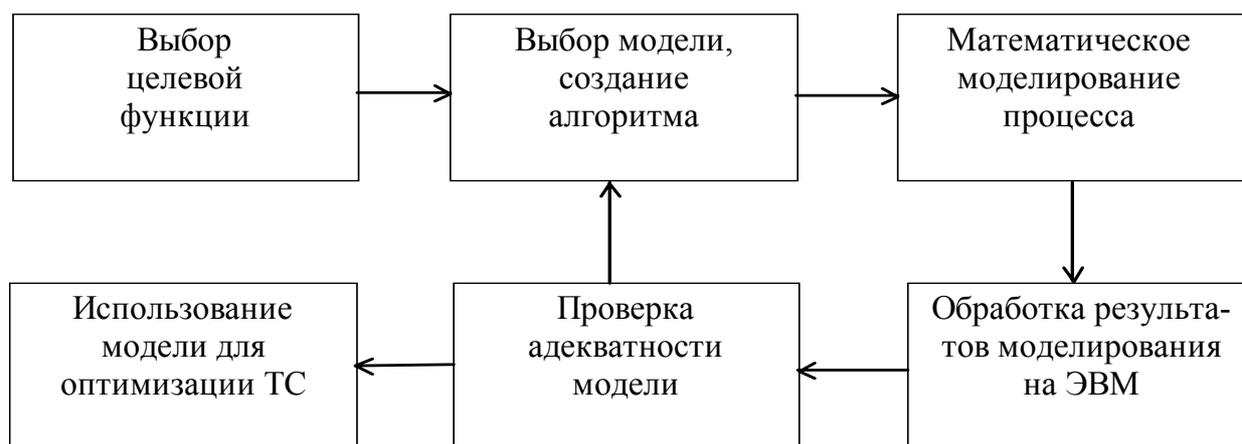


Рисунок 5.2 – Стратегия построения математической модели

Поиск оптимального решения всегда предполагает построение математической модели, т. е. представление в математической форме взаимосвязи целевой функции с параметрами задачи.

Решается данная задача методами математического программирования.

Под математическим программированием понимается совокупность методов решения задач оптимизации, связанных с нахождением наилучшего (с точки зрения выбранных критериев качества) значения некоторой целевой функции  $F(X)$  многих переменных  $X = \{x_1, x_2, \dots, x_n\}$  на заданном множестве точек  $L(X)$  евклидова пространства  $E^n$ , а также плана  $X^* = \{x^*_1, x^*_2, \dots, x^*_n\}$ ,  $X^* \in L(X)$ , при котором достигается соответствующее оптимальное значение  $F(X^*)$ .

#### Основные классы задач математического программирования:

1. Линейного программирования
2. Нелинейного программирования
3. Целочисленного программирования
4. Динамического программирования

#### Оптимизация функциональных математических моделей

При постановке задачи оптимизации в САПР необходимо преобразовать физические представления о назначении и степень сложности объекта в математическую формулировку экстремальной задачи с формулировкой критериев оптимальности.

Основу критерия оптимальности составляет целевая функция  $F$ .

Целевую функцию системы можно представить в следующем виде:

$$F = \sum \alpha_S \cdot X_S, \quad (5.1)$$

где  $\alpha_S$  - коэффициент влияния;  $X_S$  – относительное значение критериев.

Тогда оптимизация заключается в максимизации либо минимизации целевой функции.

При проведении оптимизации удобно геометрически представлять целевую функцию  $F(X)$ . При этом множество векторов  $X$  рассматривается как про-

пространство управляющих параметров.

Оптимальным вектором  $\vec{X}$  является точка, в которой целевая функция  $F(X)$  достигает  $\max$  при максимизации, либо  $\min$  при минимизации значения.

В пространстве управляемых параметров выделяют гиперплоскости равного уровня.

В точках конкретной гиперплоскости равного уровня целевая функция равна некоторой постоянной величине. Уравнение гиперплоскости:

$$F(X) = a,$$

где  $a = \text{const}$

В случае двумерного случая гиперплоскости выражаются в линии равного выхода.

Кроме целевой функции и управляемых параметров в постановку задачи оптимизации могут входить ограничения типа равенств:

$$\xi\left(\vec{X}\right) = 0,$$

ограничения типа неравенств:

$$\varphi\left(\vec{X}\right) > 0 \dots$$

Частным случаем ограничений типа неравенств являются прямые ограничения на управляемые параметры, имеющие вид:

$$X_{i \min} \leq X \leq X_{i \max},$$

где  $X_{i \min}$ ,  $X_{i \max}$  – предельно допустимые физические, технологические или выбираемые по конструктивной целесообразности значения параметра  $x_i$ .

Область пространства управляемых параметров, в которой выполняются заданные ограничения называется допустимой областью.

При наличии ограничений задача оптимизации есть задача условной оптимизации, при отсутствии – безусловной.

Точка, характеризующаяся  $\max$  или  $\min$  значением целевой функции при выполнении всех ограничений называется условным максимумом (минимумом). При отсутствии ограничений – экстремум безусловный.

Формулировка задачи поиска экстремума целевой функции  $F(X)$  в области допустимых значений  $X$ , которые имеют ограничения

$$\xi\left(\vec{X}\right) = 0, \varphi\left(\vec{X}\right) > 0$$

есть задача математического программирования.

Виды задач:

- Если хотя бы одна из функций  $F$ ,  $\xi$ ,  $\varphi$  - нелинейная, то эта задача не-

линейного программирования.

– Если все или часть параметров  $X$  являются дискретными величинами, то эта задача дискретного или частично-дискретного программирования.

– Если линейны все функции – задача линейного программирования.

Дискретное программирование называется целочисленным если управляемые параметры  $X$  принадлежат множеству целых чисел.

### ОСНОВНЫЕ КРИТЕРИИ ОПТИМАЛЬНОСТИ

1. **Частный критерий** – когда в качестве целевой функции выбирается наиболее важный выходной параметр.

2. **Максиминный критерий** – обобщенный критерий. При этом критерии вводят оценки степени выполнения каждого из условий работоспособности.

3. **Статистический критерий оптимизации.**

Целевая функция – вероятность выполнения всех заданных условий работоспособности.

Предполагают проведение статистических испытаний методами Монте-Карло, граничных испытаний или исследования образца по методу наихудшего случая.

### СПОСОБЫ ПОИСКА ОПТИМАЛЬНЫХ РЕШЕНИЙ

*На основе перебора путей на дереве решений.*

Некая подзадача  $Q_0$  разбивается на ряд подзадач  $Q_1, Q_2, \dots, Q_n$ . После разбиения подзадачи легче решаются, либо имеют меньшую размерность.

В свою очередь, каждая подзадача может разбиваться далее...

Оптимальное решение находится на основе перебора путей на дереве решений. Задачи такого типа называются **поисковыми**.

Процесс поиска связан с исследованием графовых структур.

Существует три основных вида поиска:

1. **Поиск с возвратением,**
2. **Поиск в глубину,**
3. **Поиск в ширину.**

**Поиск с возвратением** заключается в расширении частного решения на каждом шаге поиска. Если расширение частичного текущего решения невозможно – происходит возврат к предыдущему частичному решению и предпринимаются попытки снова его продолжить. Процесс идет до тех пор, пока это возможно.

По времени решения данный алгоритм относят к классу экспоненциальных.

**Поиск в глубину:** идея метода – необходимо в каждой исследуемой вершине дерева выбрать один из возможных путей и исследовать его до конца. Другие существующие пути решений при этом не рассматриваются до тех пор, пока имеется возможность получить конечный результат, исследуя выбранное направление.

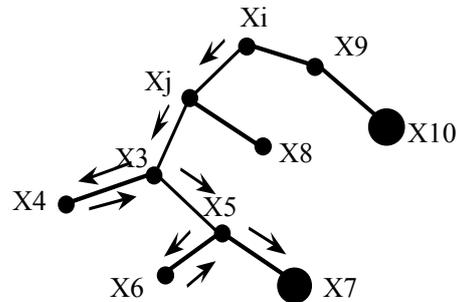
**Пример.**

Алгоритм просмотра графа  $G$  на основе поиска в глубину:

1. Выбираем произвольно начальную вершину  $X_i$ .
2. Выбираем ребро  $(X_i X_j)$ , которое инцидентно вершине  $X_i$  и просматривает вершину  $X_j$ .
3. Если в вершине  $X_j$  нет решения, то выбираем следующую вершину графа, инцидентную текущей вершине.
4. Просматриваем все пути, начинающиеся в вершине. Если решений нет, то возвращаемся назад к вершине  $X_j$  и рассматриваем очередное ребро. Если решение найдено – вычисления прекращаются.
5. Если ни на одном пути из вершины  $X_j$  нет решения, то выбирается новая вершина  $X_j$  инцидентная  $X_i$  и далее с п. 2.

*Недостаток:* при исследовании дерева решений с большой вероятностью можно пройти мимо той ветви, на которой раньше всего появляется окончательное решение.

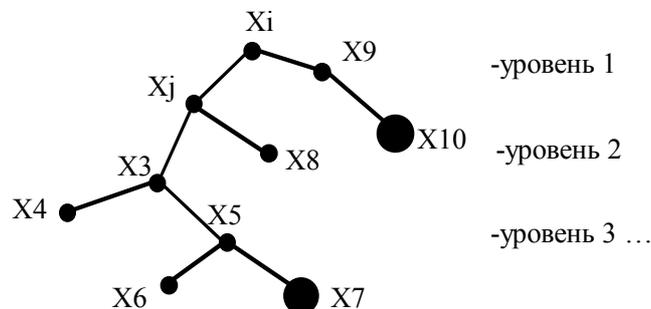
Пример:



● - решение задачи  
Рисунок 5.3

**Поиск в ширину.** При этом методе вначале исследуются задачи уровня 1. Затем задачи уровня 2

Пример:



● - решение задачи  
Рисунок 5.4

## 5.2 Структурно-логические модели ТП. Общие сведения

Согласно ГОСТ 24.416-83, структурно-логические модели при технологическом проектировании подразделяют на:

1. табличные,
2. сетевые и
3. перестановочные.

Причем они определяются строками булевой матрицы:

$F_G$	$F_n$	$F_\lambda$	$F_a$	
1	1	1	1	$S_1$
1	1	1	0	$S_2$
1	1	0	1	$S_3$
1	1	0	0	$S_4$
1	0	1	0	$S_5$
1	0	0	0	$S_6$
0	1	1	1	$S_7$
0	1	1	0	$S_8$
0	1	0	1	$S_9$
0	1	0	0	$S_{10}$
0	0	1	0	$S_{11}$
0	0	0	0	$S_{12}$

$S_i$  – класс моделей, который характеризуется набором контуров  $F_i$  (условия).

$F_G$  – условие, определяющее маршрут операции или переходов технологического процесса:  $A_k (F_G=1)$ , маршрут, будучи представленным в виде графа, будет являться простой цепью ( $F_G=0$  - в противном случае).

$F_n$  - условие, определяющее маршрут операции и переходов технологического процесса. Если  $F_n = 1$ , то количество элементов в маршруте постоянное.  $F_n = 0$  – иначе.

$F_\lambda$  – условие, учитывающее отношение порядка элементов (операций переходов в технологическом процессе). Если  $F_\lambda = 1$ , то отношение порядка не меняется,  $F_\lambda = 0$  – иначе.

$F_a$  – условие, учитывающее состав операций и переходов технологического процесса. Если  $F_a = 1$ , то состав операций одинаковый, если  $F_a = 0$  – иначе.

Контур  $F_i$  характеризуют разные стороны унификации технологического процесса:

- при  $F_a = 1$  унифицируется *состав технологических операций*.
- при  $F_n = 1$  унифицируется *количество операций*.
- при  $F_\lambda = 1$  унифицируется *последовательность реализации операции*.

- при  $F_G = 1$  унифицируется *характер смежности операций в технологическом процессе*.

**Структурно-логические модели подразделяются на:**

1. **Табличная модель** – описывает одну конкретную структуру технологического процесса (это *модели класса  $S_1$* ). В табличной модели *одному набору условий или контуров соответствует единственный вариант технологического процесса*. В связи с этим эти модели *используют для описания стандартных и типовых проектных решений*.

2. **Сетевая модель** – описывает множество структур технологического процесса, отличающихся количеством и составом элементов структуры при неизменном отношении порядка (это *модели классов  $S_2, S_5, S_7, S_8, S_{11}$* ). В сетевой модели структура элементов описывается ориентировочным графом, который не имеет ориентировочных циклов.

3. **Перестановочная модель** – описывает множество структур технологического процесса, отличающихся количеством и составом элементов структуры при изменении отношения порядка (это *модели классов  $S_3, S_4, S_6, S_9, S_{10}, S_{12}$* ). Отношение порядка в перестановочных моделях задается с помощью графа, содержащего перестановочные циклы.

Эти модели называются *структурно-логическими*, так как они *задают состав и взаимосвязь операций технологического процесса*.

Они представляются в виде графа, который определяет состав и последовательность (маршрут) выполнения этапов, операций, переходов при выполнении обработки или сборки изделия.

Вершины графа соответствуют элементам технологического процесса (этапам, операциям, переходам), а рёбра или дуги графа характеризуют последовательность элементов технологического процесса.

Таким образом, исходя из этих определений можно сделать вывод, что модель относится к той или иной группе в зависимости от условий налагаемых на ТП.

При описании взаимосвязи контуров изделий технологической системы используют

**дизъюнктивную и**

**конъюнктивную** форму представления этих связей.

При *дизъюнктивной форме* исходное состояние контуров  $F_i(A)$  изделия обозначается равным нулю. Оконечное состояние после реализации соответствующей операции  $F_i(A)=1$ .

Т. е. *Дизъюнктивная форма* описывает переход контура  $F_i$  в реализованное состояние, под воздействием конкретной технологической операции  $\tau_k$  (конечный контур, равный единице), т.е

$$F_i(\tau_k)=1.$$

Состав контуров изделия после выполнения  $k$ -й операции в дизъюнктивной форме можно записать так:

$$F(A)_i = F(A)_{k-1} \cup F(\tau_k)$$

При **конъюнктивной форме** исходные состояния контуров изделия  $F_i(A)=1$ , контура, подлежащего реализации  $F_i(A)$  принимает единичное значение  $F_i(A)=1$ . Состояние контура  $F_i(A_k)$  технологической операции, участвующей в реализации  $F_i(A)$  также обозначается  $F_i(\tau_k)=1$ .

Матрица контуров операций в этом случае называется конъюнктивной. Состав контуров изделия  $F(A_k)$ , реализуемых при участии  $k$ -ой операции определяется по формуле:

$$F(A)_k = F(A) \cap F(\tau_k)$$

### 5.3 Табличная модель ТП

Пусть дана группа деталей  $a_1, a_2, a_3$ , описание которых объединено в одно множество  $A$ .

$$a_1 \cup a_2 \cup a_3 = A$$

Детали имеют форму тел вращения и обрабатываются на токарном автомате с помощью табличной модели (рисунок 5.5). Каждая деталь имеет поверхности с определенными свойствами.

Устанавливается последовательность обработки этих деталей.

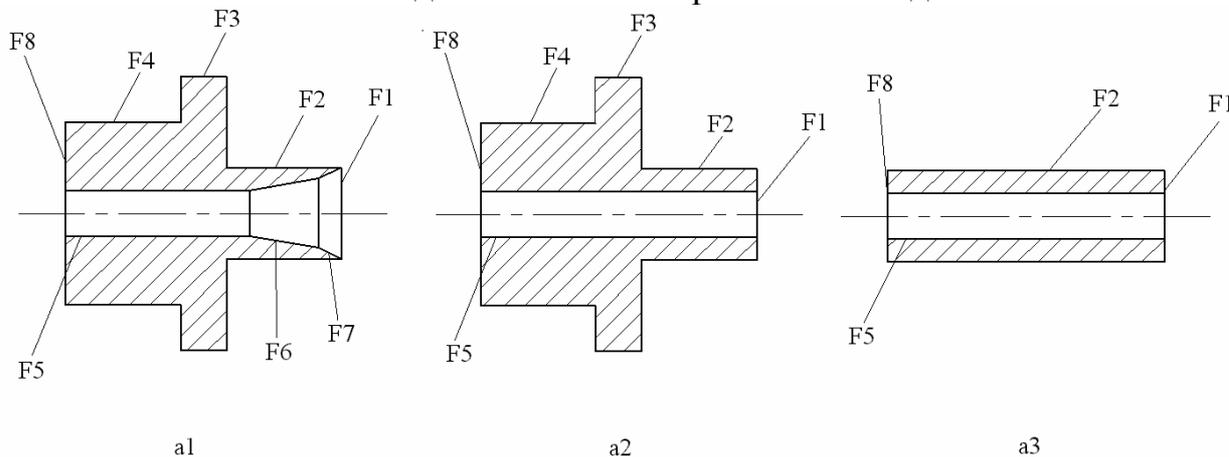


Рисунок 5.5 – Чертежи деталей – тел вращения

Для реализации свойств изделия используются технологические операции и переходы  $\tau_k$ , в нашем случае:

$$F(a_1) = \{F_1 \dots F_8\},$$

$$F(a_2) = \{F_1, \dots, F_5, F_8\},$$

$$F(a_3) = \{F_1, F_2, F_5, F_8\},$$

$\tau_1$  – подрезка торца;

$\tau_{2,3,4}$  – точение цилиндрических поверхностей;

$\tau_5$  – сверление;

$\tau_6$  – коническое растачивание;

$\tau_7$  – зенкование;

$\tau_8$  – отрезка.

Табличные модели представляются в виде матриц и графов:

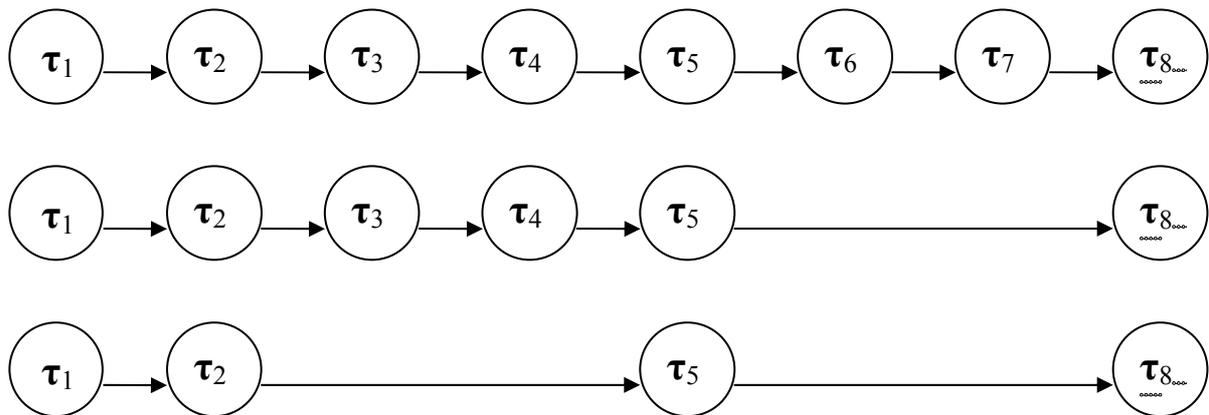


Рисунок 5.6 – Графы взаимосвязей переходов для деталей  $a_1, a_2, a_3$

### Дизъюнктивная матрица

		$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$
Подрезка торца	$\tau_1$	<b>1</b>							
Точение цил. пов-тей	$\tau_2$		<b>1</b>						
Точение цил. пов-тей	$\tau_3$			<b>1</b>					
Точение цил. пов-тей	$\tau_4$				<b>1</b>				
Сверление	$\tau_5$					<b>1</b>			
Растачивание	$\tau_6$						<b>1</b>		
Зенкерование	$\tau_7$							<b>1</b>	
Отрезка	$\tau_8$								<b>1</b>

### Конъюнктивная матрица

	$F(a_1)$	$F(a_2)$	$F(a_3)$
$\tau_1$	<b>1</b>	<b>1</b>	<b>1</b>
$\tau_2$	<b>1</b>	<b>1</b>	<b>1</b>
$\tau_3$	<b>1</b>	<b>1</b>	
$\tau_4$	<b>1</b>	<b>1</b>	
$\tau_5$	<b>1</b>	<b>1</b>	<b>1</b>
$\tau_6$	<b>1</b>		
$\tau_7$	<b>1</b>		
$\tau_8$	<b>1</b>	<b>1</b>	<b>1</b>

### ***Алгоритм проектирования с использованием дизъюнктивной модели***

- 1) деталь описывается набором свойств  $F_i$  и на основании этого описания
- 2) проверяется достаточность контуров модели для реализации детали.

$$F(A) \subseteq F(\tau) \quad (\text{контур } F(A) \text{ содержится в } F(\tau))$$

3) рассматривают очередную строку матрицы и проверяют, участвует она в реализации контуров изделия, т.е. отыскивается «1» в строке матрицы. Это выражается следующим образом:

$$F(A_k)^1 = F(A)_{k-1} \cap F(\tau_k)$$

*(перемножают строку  $\tau_k$  на столбец  $F_{k-1}$ )*

4) если  $F(A_k)^1 \neq 0$ , то операция  $\tau_k$  включается в технологический процесс, если  $F(A_k)^1 = 0$ , то операция не включается и происходит анализ следующей строки.

*Необходимо отметить, что порядок следования операций в этом случае не меняется.*

5) Определяются все контуры, над которыми выполняются операции и проверяются, все ли контуры из заданной системы обработаны.

6) После включения операции в ТП проверяется, все ли контуры изделия реализованы. Если да – проектирование завершено, если нет – возвращаемся к п. 3.

### ***Алгоритм проектирования с использованием конъюнктивной модели***

1. Подготовка исходных данных описания контуров (свойств) деталей  $F(A)$ .

2. Проверяют составы контуров модели  $F(\tau)$  для изготовления данного изделия  $F(A) \subseteq F(\tau)$ . Если достаточно - переходят к дальнейшему анализу, иначе проектирование прекращается.

3. Рассматривают  $k$ -ую строку матрицы контуров и проверяют, участвует ли данная операция в изготовлении изделия  $F(A)_k = F(A) \cap F(\tau_k)$

4. по формуле  $F(A) = F(A)_k \cap F(A)$  проверяется все ли контуры, которые реализуются с помощью  $\tau_k$ , включены в модель. При положительных ответах на оба вопроса операция включается в состав ТП, иначе – нет.

5. Проводится проверка, все ли операции просмотрены (т.е. строки матрицы отношений контуров и операций). Если да – конец проектирования. Иначе – переход к п. 3.

В табличных моделях - постоянный порядок операций, определённый матрицей отношения операций и контуров.

#### ***Свойства моделей класса $S_1$ (табличных):***

- 1) нет циклов;
- 2) количество элементов постоянно;
- 3) порядок постоянен;
- 4) состав операций неизменен.

## 5.4 Сетевая модель ТП

Модели  $S_2, S_5, S_7, S_8, S_{11}$  – сетевые модели, которые отличаются количеством и составом операций при неизменном отношении порядка  $F(\lambda)$ . В модели может содержаться несколько вариантов Ак проектируемого ТП, однако, во всех будет неизменным отношение порядка.

Сетевая модель включает в себя матрицу свойств детали, описания логических отношений между свойствами и граф  $G(T, C)$  взаимосвязи операторов ( $T = \{ \tau_1, \tau_2 \dots \tau_n \}$  – операции;  $C = \{ C_1, C_2 \dots C_n \}$  – дуги графа) по возможной их последовательности.

Рассмотрим модель сетевую  $S_5$  (отсутствие циклов  $F_G = 1$ , неизменное отношение порядок  $F_\lambda = 1$ ) изготовления зубчатого колеса (рисунок 5.7).

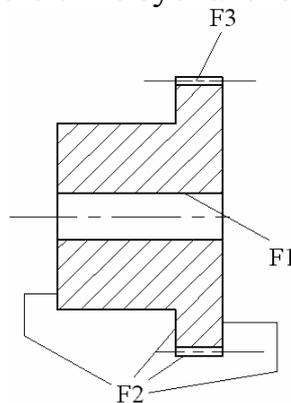


Рисунок 5.7 – Зубчатое колесо

### Множество $T$ :

- $\tau_1$  – получение заготовки;
- $\tau_2$  – сверление базового отверстия;
- $\tau_3$  – черновое обтачивание контура зубчатого колеса;
- $\tau_4$  – чистовое обтачивание контура зубчатого колеса;
- $\tau_5$  – черновая маркировка зубчатого колеса;
- $\tau_6$  – чистовая черновая маркировка зубчатого колеса;
- $\tau_7$  – термообработка;
- $\tau_8$  – отделка базового отверстия;
- $\tau_9$  – шлифование зубчатого профиля;
- $\tau_{10}$  – притирка зубчатого профиля;
- $\tau_{11}$  – мойка;
- $\tau_{12}$  – контроль.

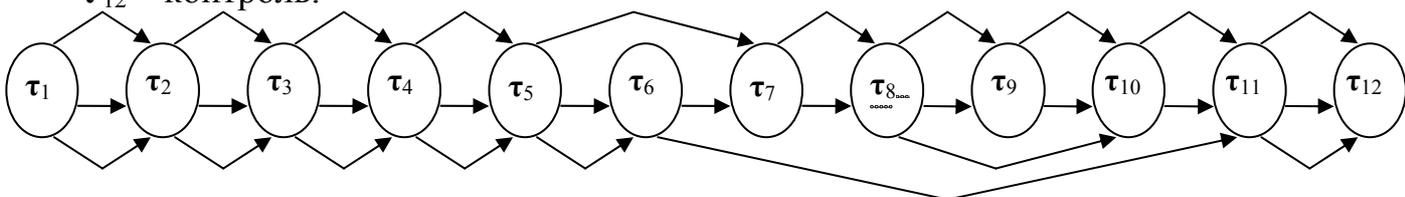


Рисунок 5.8 – Граф взаимосвязи операторов по возможной последовательности их реализации

Алгоритм проектирования по сетевым моделям включает:

- операторы построения путей на графе  $G=(T, C)$ ,
- операторы проектирования по табличной модели для каждого пути и
- операторы выбора оптимальных проектных решений.

В рассматриваемом примере max ТП состоит из 12 технологических операций (на графе). Представлено 4 возможных варианта технологического процесса и составления матрицы отношений контуров и операций.

1 путь – все операции.

2 путь - исключена чистовая нарезка зубчатого профиля ( $\tau_6$ ).

3 путь – исключены операции  $\tau_7 - \tau_9$ .

4 путь – исключена  $\tau_9$  шлифование.

Выделив один путь, *проектируем ТП по табличной модели*,

- затем просматриваем все пути и

- находим оптимальное решение.

Матрица взаимосвязи операции и контуров:

	$F_i \backslash \tau_i$	$F_1$	$F_2$	$F_3$
Получение заготовки	$\tau_1$	1	1	
Протягивание базового отверстия	$\tau_2$	1		
Черновое обтачивание контура зубчатого колеса	$\tau_3$		1	
Черновое обтачивание контура зубчатого колеса	$\tau_4$		1	
Черновая нарезка зубчатого профиля	$\tau_5$			1
Черновая нарезка зубчатого профиля	$\tau_6$			1
Термообработка	$\tau_7$	1	1	1
Отделка базового отверстия	$\tau_8$	1		
Шлифование базового отверстия	$\tau_9$			1
Притирка зубчатого профиля	$\tau_{10}$			1
Мойка	$\tau_{11}$	1	1	1
Контроль	$\tau_{12}$	1	1	1

### 5.5 Перестановочная модель ТП

Модели  $S_3, S_4, S_6, S_9, S_{10}, S_{12}$  – перестановочные. Эти модели имеют различный порядок операций. Отношение порядка между элементами задаётся с помощью графа, содержащего ориентированные циклы. Получается n вариантов ТП и в каждом варианте различается отношение порядка.

При проектировании по перестановочным моделям необходимо:

- а) подготовить математические данные описания детали.
- б) проверить применимость модели для проектирования ТП для данной детали.
- в) вычислить набор перестановок.

г) определить, удовлетворяют ли они ограничениям, накладываемым на реализацию ТП и включить только удовлетворяющие в состав проверяемых вариантов.

д) с помощью табличных моделей спроектировать ТП, согласно всем вычисленным путям и всем возможным наборам операций.

е) выбрать оптимальный процесс.

**Пример:** Рассмотрим модель при построении расцеховки её граф следующий (рисунок 5.9):

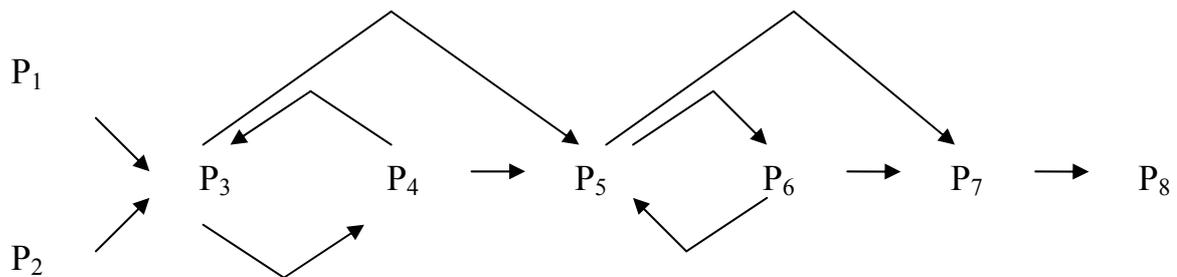


Рисунок 5.9 – Граф расцеховки изделия

Где :

P<sub>1</sub> – литейной

P<sub>2</sub> – кузнечной

P<sub>3</sub> – механической

P<sub>4</sub> – нормической

P<sub>5</sub> – механосборочной

P<sub>6</sub> – общей сборки

P<sub>7</sub> – испытаний

P<sub>8</sub> – упаковки

Для каждого изделия, очевидно, есть много вариантов прохождения цехов и изготовления деталей. Эта модель типа S<sub>10</sub> модель с переменным порядком операций.

Данная модель очень часто используется для описания технологических процессов сборки, т.к. в них нарушается отношение порядка.

### Пример.

Необходимо синтезировать ТП сборки редуктора (рисунок 5.10).

Сборочные операции ( $\tau_i$ ):

$\tau_1$  – установка вала в сборочное приспособление;

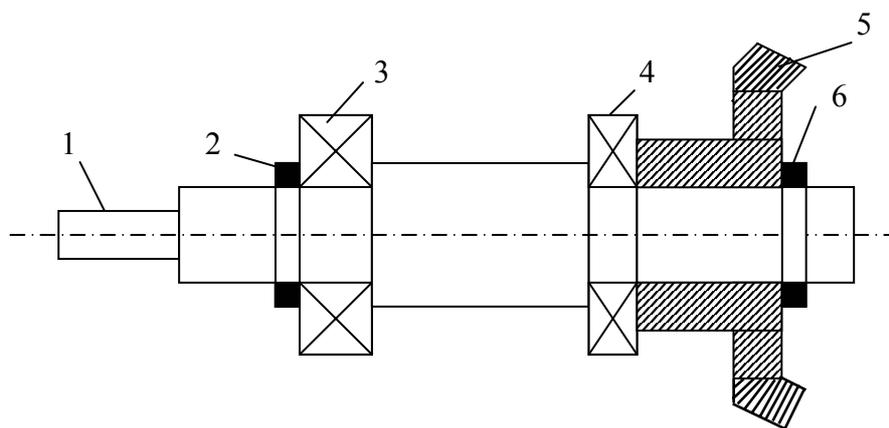
$\tau_2$  – установка стопорного кольца (2);

$\tau_3$  – посадка шарика-подшипника(3);

$\tau_4$  – посадка шарика-подшипника(4);

$\tau_5$  – установка конического колеса (5);

$\tau_6$  – установка стопорного кольца (6);



1 – вал; 2 – стопорное кольцо; 4, 3 – подшипник; 5 – коническое зубчатое колесо; 6 – стопорное кольцо

Рисунок 5.10 – Сборочный чертеж редуктора

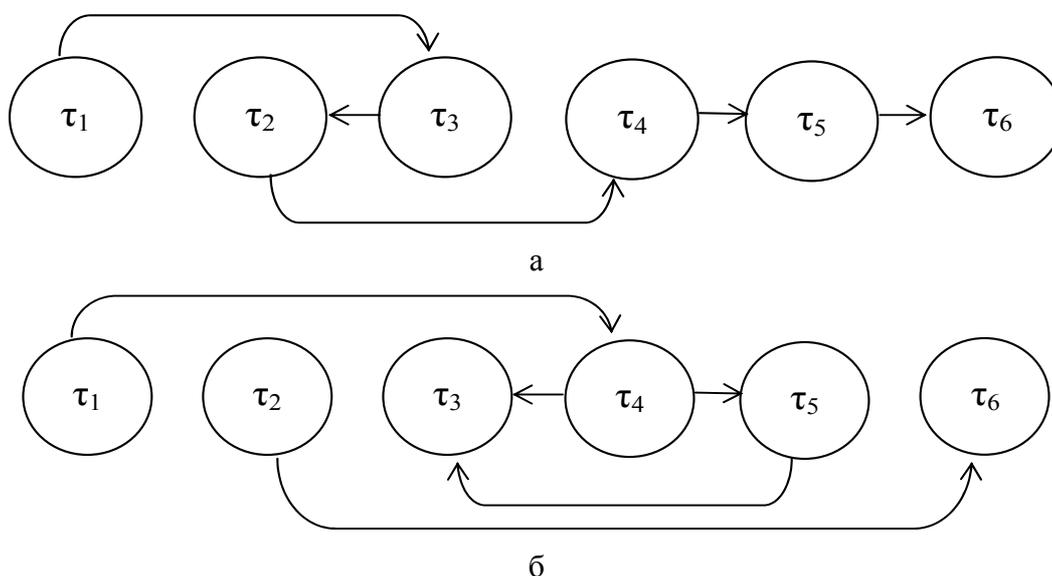


Рисунок 5.11 – Графы вариантов ТП сборки редуктора

Алгоритм проектирования технологического процесса по дизъюнктивной табличной модели:

- 1) Готовятся исходные данные о составе контуров изделия, которые должны быть реализованы в этой модели.
- 2) Проверяется достоверность состава контуров модели для изготовления конкретного изделия.
- 3) Рассматривается очередная  $k$ -я строка дизъюнктивной матрицы контуров.
- 4) Проверяется, участвует ли операция  $\tau_k$  в реализации контуров изделия. Если да, то переходим к п. 5, если нет, то возвращаемся в п. 3.
- 5) Операцию  $\tau_k$  включают в технологически процесс  $T_i$ , очередность выполнения операций  $\tau_k$  в технологическом процессе определяется очередностью включения  $\tau_k$  в  $T_i$ .
- 6) Определяется состав контуров, которые могут быть реализованы набором  $T_i$  операций.

7) Проверяется достоверность этих контуров для реализации изделия. Если да, то проектирование закончено, если нет, то переходим к п. 3.

### ***Способ построения графа вариантов маршрута сборки изделия***

1) Выбирается элемент конструкции, который называется базовой деталью и он принимается за корневую вершину графа (*чаще всего это корпус*).

2) Рассматриваются все остальные элементы конструкции кроме корневого, и проверяется выполнение условий базирования и доступа. Отобранные таким образом вершины образуют второй ярус в графе.

3) Рассматриваются остальные вершины за исключением включенных в граф и т.д.

4) Построение графа заканчивается тогда, когда для каждой ветви получены последние висячие вершины, соответствующие последним элементам сборочной единицы.

Выбор наилучшего варианта маршрута проводится по какому-то определенному критерию в зависимости от конструкторских условий (*чаще всего это временные параметры – трудоемкость изготовления*).

## **5.6 Типовые решения в САПР технологических процессов**

Особенности проектирования технологических процессов:

**1. Многовариантность проектных решений.**

**2. Слабая формализация многих проектных задач.**

Задачи формального расчетного характера при проектировании ТП:

**- расчет припусков и межпереходных размеров;**

**- расчет режимов резания;**

**- нормирование технологического процесса.**

Типовые решения – это основа формализации для решения задач неформального характера при проектировании технологических процессов с помощью ЭВМ.

### **Алгоритм выбора решений**

1) каким-либо образом описывается весь набор типовых решений,

2) описывается набор условий, при которых может быть применено каждое из них.

3) Эти данные описываются заранее в виде базы данных и вводятся в ЭВМ.

4) При разработке ТП в ЭВМ вводятся некоторые исходные данные по детали.

5) Проверяется соответствие исходных данных условиям применимости типовых решений.

### **Виды типовых решений**

1) **локальные типовые решения (ЛТР) и**

2) **полные типовые решения (ПТР).**

**Локальные типовые решения** относятся к частным технологическим задачам, определяющим лишь некоторую часть (элемент) проектируемого технологического процесса.

**Полные типовые решения** охватывают весь (полный, логически завершённый) круг решаемых задач.

Примером полного типового решения является типовой технологический процесс.

### **Иерархические уровни технологического проектирования**

1. Разработка принципиальной схемы технологического процесса (последовательность этапов укрупнённых операций).
2. Проектирование технологического маршрута сборки или обработки.
3. Разработка управляющих программ для станков и ИПУ.

В зависимости от условий изготовления объекта и его особенностей ТП делят на *единичные, типовые и групповые*.

*Единичные технологические процессы* устанавливаются на изделия одного наименования, типоразмера и исполнения, независимо от типа производства.

*Типовые технологические процессы* устанавливаются на группу изделий с общими конструктивными признаками (например, изготовление печатных плат).

*Групповые технологические процессы* устанавливаются на конструктивно и технологически схожие изделия.

### **Типовые технологические процессы (по профессору А.А.Соколовскому)**

*Идея типизации* заключается в классификации деталей по конструктивно – технологическим признакам: форме, размерам, точности и т.д.

*Конечная цель классификации* – установление принадлежности детали к определённому типу, т.е. к совокупности деталей, имеющих в данных производственных условиях **общую структуру операций и переходов**.

*Технологический процесс обработки* конкретной детали получается из типового путем исключения лишних операций и переходов обработки отсутствующих поверхностей.

*Доработка типового технологического процесса включает в себя также:*

- уточнение технологического оснащения (оборудования, приспособлений, инструментов);
- перерасчет межпереходных размеров;
- выбор (расчет) режимов резания;
- выбор и расчет норм времени.

## **Групповые технологические процессы (по профессору С.П.Митрофанову)**

1) Классификация, заканчивающаяся формированием группы, т.е. совокупности деталей, характеризующихся общностью оборудования и оснащения, необходимых для обработки детали в целом или отдельных ее поверхностей.

2) Уточнения общих поверхностей с комплексной деталью группы и выбора из группового технологического процесса только тех операций и переходов, которые необходимы для обработки поверхностей заданной детали.

3) Доработка индивидуального технологического процесса.

## 6 МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

В зависимости от степени полноты реализации синтеза и анализа выделяют:

1. Метод прямого проектирования (документированного).
2. Метод анализа (адресации, аналога).
3. Метод синтеза.
4. Комбинированные методы.

### 6.1 Метод прямого проектирования

*Суть:* подготовка проектного документа (технологической карты) возлагается на самого пользователя, выбирающего типовые решения различного уровня из базы данных в диалоговом режиме.

#### *Алгоритм проектирования*

1) Заранее создается и заполняется технологическая база данных, включающая в себя информацию об имеющихся на предприятии заготовках, станках, приспособлениях, инструментах и т.д.

2) Пользователю представляются меню на разных уровнях проектирования для выбора заготовок, операций, станков, приспособлений, переходов, инструментов и т.д.

Выбранная информация автоматически заносится в графы и строки *шаблона технологической карты*.

3) В режиме редактирования информация при необходимости может редактироваться,

4) Печать ТД в форме, предусмотренной соответствующим ГОСТом.

### 6.2 Метод анализа

В его основе лежат *полные типовые решения*.

*Суть:* структура индивидуального технологического процесса не создается заново. Она определяется в соответствии с составом и структурой одного из унифицированных технологических процессов, т.е. соответствующего типового или группового технологического процесса.

Это осуществляется путем **анализа необходимости** каждой операции и перехода с последующим уточнением всех решений на уровнях декомпозиции *«сверху – вниз»*.

Метод воплощает идею *«от общего к частному»*.

#### *Алгоритм проектирования*

- 1) ввод описания чертежа детали
- 2) определение конструктивно – технологического кода детали

- 3) поиск по коду в базе данных приемлемого унифицированного (типового или группового) технологического процесса
- 4) анализ структуры ТП
- 5) доработка ТП в соответствии с описанием чертежа детали
- 6) оформление индивидуального технологического процесса.

**Особенность применения метода** – большая подготовительная работа.

Из множества деталей заводской номенклатуры формируются группы, имеющие общие конструктивно – технологические признаки, способы обработки.

**Два подхода:**

1. В каждой группе выбирается деталь – представитель и для нее разрабатывается **типовой технологический процесс**.

- Все типовые технологические процессы для всех групп деталей заносятся в ЭВМ.

- При разработке индивидуального технологического процесса из типового технологического процесса, как правило, исключаются лишние операции и переходы.

- Иногда, что гораздо реже, недостающие операции и переходы могут добавляться в режиме ручного редактирования технологического процесса.

- Далее уточняется оборудование, технологическая оснастка, выбираются или рассчитываются режимы резания, рассчитываются нормы времени.

2. Для каждой группы формируется обобщенная модель всех деталей – комплексная деталь. Она включает все многообразие поверхностей рассматриваемой группы.

- Для комплексной детали разрабатывается унифицированный (**групповой технологический процесс**). Он заведомо является избыточным, т.е. содержит операции и переходы по обработке всех деталей группы. Разработка индивидуального технологического процесса заключается в анализе необходимости включения в него операций и переходов из соответствующего группового технологического процесса. Или, другими словами, из группового технологического процесса исключаются лишние операции и переходы (см. рис.8.3). Затем выполняется, как и в первом случае, так называемая параметрическая настройка: уточнение оборудования, технологической оснастки, выбор или расчет режимов резания и т.д.

**Метод анализа** является основным методом проектирования технологических процессов при эксплуатации гибких производственных систем. Его применение дает наибольший эффект при внедрении на производстве групповых и типовых технологических процессов.

**Достоинства:**

- 1) метод не нарушает существующей специализации и традиций производственных подразделений,
- 2) упрощает процесс проектирования,

3) не требует трудноформализуемых процедур синтеза новых технологических процессов.

### **6.3 Метод синтеза в САПР технологических процессов**

В основе метода синтеза лежат *локальные типовые решения*.

Алгоритмы построения САПР на основе метода синтеза существенно отличаются друг от друга.

*Причины:*

1. Процедуры разработки (синтеза) технологических процессов относятся к разряду трудноформализуемых.

2. Ряд САПР, построенных по методу синтеза, ориентированы на проектирование ТП изготовления деталей определенного класса (например, «тел вращения»).

3. С целью исключения циклов при разработке технологии и обеспечения линейной стратегии проектирования некоторые разработчики САПР отошли от классической схемы проектирования ТП «маршрут–операция–переход» и т.д.

#### **Реализация линейной стратегии проектирования в САПР технологических процессов**

Версия стратегии разработана на кафедре «Технология машиностроения» Ярославского государственного технического университета.

*Алгоритм проектирования*

**1. Ввод описания чертежа детали.**

2. Синтез маршрутов (планов) обработки для всех поверхностей детали.

3. Синтез принципиальной схемы технологического процесса.

4. Синтез маршрута обработки детали.

5. Синтез состава и структуры операций технологического процесса.

6. Доработка технологического процесса (расчет режимов резания, нормирование).

**7. Оформление документации.**

Ввод описания чертежа детали и оформление документации являются **общими этапами для всех методик проектирования ТП в САПР.**

#### **Синтез маршрутов обработки поверхностей**

*Маршрут обработки поверхности* (МОП) – это последовательность методов (видов, переходов одного метода) обработки, необходимых для достижения требуемых чертежом детали параметров поверхности.

Таковыми параметрами являются:

- геометрический тип поверхности;
- точность размера;
- шероховатость;

- вид термообработки и т.д.

### **Синтез принципиальной схемы технологического процесса**

В базе данных хранится **принципиальная схема**, разбивающая будущий технологический процесс на последовательность отдельных **этапов обработки**.

Наличие принципиальной схемы позволяет вести проектирование технологического процесса **в порядке, обратном изготовлению детали**, т.е. от заключительных этапов с известных из чертежа параметров детали, к черновым этапам, заканчивая выбором размеров и формы заготовки.

Принципиальная схема технологического процесса построена на основе анализа обработки деталей различных классов с учетом возможных комбинаций термической и последующей механической обработки.

**Этап** – часть технологического процесса обработки детали, включающая однородные по достигаемым параметрам методы обработки различных поверхностей и детали в целом.

К одному этапу относятся, например, тонкое фрезерование и тонкое (торцовое) точение, т.к. оба этих метода обеспечивают одинаковые параметры точности и шероховатости поверхности.

С помощью условий принципиальной схемы анализируется необходимость при обработке детали каждого из 13 этапов, и выявляются те переходы из выбранных ранее в каждом МОП, которые должны быть выполнены на данном этапе.

Отнесение переходов МОП и самих промежуточных поверхностей к этапам принципиальной схемы технологического процесса производится путем сравнения параметров поверхности заготовки и детали, обеспечиваемых переходом МОП, с параметрами, характеризующими один из 13 этапов схемы (выбирается самый близкий этап).

В результате распределения переходов по этапам каждый этап будет содержать переходы одного или разных методов обработки с одинаковыми (или близкими) параметрами точности, шероховатости и т.д.

### **Синтез маршрута обработки детали**

*Исходные данные:*

1. Полученная ранее структура принципиальной схемы этапов технологического процесса.
2. Сформированный набор методов – переходов одного или разных МОП в каждом этапе.

Каждый переход записан в памяти компьютера в промежуточные массивы в **виде многоуровневого кода**. Этот код состоит из номера получаемой поверхности и кода метода обработки.

Например, код **31101** означает:

3 – третья поверхность детали;

1 – первое промежуточное состояние этой (третьей поверхности);

101 – метод обработки (получистовое точение), в результате которого дан-

ное промежуточное состояние достигнуто.

***Код метода обработки определяет код применяемого оборудования.***

Переходы одного метода обработки, коды которых имеют одинаковый код оборудования (или метода обработки), образуют укрупненную операцию этапа.

При формировании маршрута обработки детали решаются следующие **задачи**:

1. Определяется состав операций.
2. Укрупненные операции дифференцируются на простые.
3. Формируется последовательность операций на каждом этапе.
4. Выбирается тип оборудования для каждой операции.

***Способы решения:***

1. Расчленение исходного набора переходов этапа на укрупненные операции выполняются программным модулем, представляющим собой процедуру сортировки кодов отобранных в этап переходов по признаку типа оборудования.

2. Дифференциация операций осуществляется путем определения их рационального состава и последовательности обработки. Алгоритмы основаны на анализе отношений между поверхностями детали:

- **наложения**, когда одна поверхность расположена на другой и поэтому не может быть обработана раньше;

- **точности взаимного расположения**, когда в первую очередь должна быть обработана базовая поверхность, и только затем поверхности, точности взаимного расположения которых заданы относительно данной базовой поверхности и т.д.

3. Формирование последовательности операций выполняется путем выявления признаков технологической совместимости и предшествования.

Две операции попарно совместимы, если состояние детали на выходе одной операции может быть исходной для другой. Так фрезерованию шпоночной канавки должна предшествовать токарная обработка цилиндрической поверхности, сверлению центрального отверстия – фрезерование торца и т.д.

***Источником информации для выбора оборудования являются технологические признаки кода перехода.***

Конкретная модель станка определяется по таблицам соответствий или обращением к базе данных с учетом габаритных размеров детали (сопоставляется с размерами рабочей зоны станка), требований точности, величины партии и других факторов.

**Синтез состава и структуры операций**

Задача данного уровня декомпозиции проектирования состоит в том, чтобы определить оптимальную последовательность переходов, рациональную форму, окончательные и промежуточные размеры заготовки.

Поиск (синтез) оптимальной операции включает в себя две задачи:

1. **Структурную оптимизацию** – создание оптимальной структуры опе-

рации (элементов системы СПИД, порядка выполнения переходов).

2. **Параметрическую оптимизацию** – определение оптимальных параметров (припусков и межпереходных размеров, режимов резания).

В основе решения задач *структурной оптимизации* заложен перебор конечного множества вариантов, состоящий из трех этапов:

1. Собственно синтез очередного варианта.
2. Анализ (оценка) варианта.
3. Принятие решения о замене ранее выбранного варианта на новый вариант или о прекращении синтеза новых вариантов.

Для оценки уровня создаваемых вариантов вводится целевая функция, выражающая качество варианта. Она формируется на основе критерия оптимальности, в качестве которого могут выступать:

- технологическая себестоимость детали (операции);
- производительность операции и т.д.

Примеры шагов алгоритма :

- выбор вариантов технологических баз и схемы базирования (одна из трудноформализуемых процедур, выполняется, как правило, в диалоговом режиме);
- определение последовательности обработки поверхностей и т.д.

### **Доработка технологического процесса**

*На данном этапе определяют режимы резания и производят нормирование технологического процесса.*

Способы определения:

1. Выбор нормативных значений режимов резания из базы данных по известному набору исходных данных (точности, шероховатости, материала заготовки и т.д.).

2. Решение задачи параметрической оптимизации режимов резания методом линейного программирования для случая использования степенных зависимостей (в историческом плане – это один из первых примеров использования ЭВМ в технологическом проектировании).

3. Решение задачи параметрической оптимизации режимов резания методами нелинейного программирования (в случае прямого использования нелинейных зависимостей для ограничений и целевых функций).

### **Оформление документации**

Вся рассчитанная и хранящаяся в разных промежуточных массивах технологическая информация сводится в единый документ – маршрутную или операционную технологическую карту.

Шаблоны этих карт содержатся в составе постоянной информации базы данных.

## 6.4 Оптимизация технологических процессов в САПР ТП

Технологический процесс называется *оптимальным*, если он обеспечивает:

1. Выполнение системы ограничений, отражающих условия протекания ТП и требования, предъявляемые к нему и детали.
2. Экстремум целевой функции.

*ТП, оптимальный по одному критерию, может быть далеко не оптимальным по другому.*

**Важным** является правильный выбор критерия оптимальности.

Наиболее часто используются следующие критерии оптимальности ТП:

1. Штучное время –  $T_{шт}$  (целевая функция  $T_{шт} \rightarrow \min$ ).
2. Производительность  $Q$  (целевая функция  $Q \rightarrow \max$ ).
3. Себестоимость детали  $C$  (целевая функция  $C \rightarrow \min$ ).

Для *постановки задачи оптимизации ТП* необходимо сформировать *математическую модель* процесса обработки детали (сборки изделия).

*Математическая модель* должна включать в себя:

1. Критерий (критерии) оптимальности ТП.
2. Целевую функцию.
3. Систему ограничений.
4. Четко определенные входные, выходные и внутренние параметры.
5. Управляемый (варьируемый) параметр или управляемые (варьируемые) параметры, которые выделяются из числа внутренних параметров.

**Затем** необходимо определить (выбрать, разработать) **метод решения задачи оптимизации**.

**Виды оптимизации ТП:**

1. Структурная.
2. Параметрическая.
3. Структурно – параметрическая.

**Структурная оптимизация** – это определение оптимальной структуры ТП (вида заготовки, технологического маршрута, модели оборудования, типоразмера инструмента и т.д.).

**Параметрическая оптимизация** ТП заключается в расчете оптимальных припусков и межпереходных размеров, режимов резания и т.д.

**Структурно – параметрическая** оптимизация представляет собой комбинацию двух первых.

*Отличие между параметризациями состоит в сущности оптимизируемых параметров.*

При **структурной оптимизации оптимизируемые параметры** по своей природе являются **неупорядоченными переменными**. В структурной же опти-

мизации эти параметры *не являются по существу числовыми*.

В параметрической оптимизации параметры представляют собой *переменные, для которых существует понятие больше или меньше и которые естественным образом могут быть размещены в координатной системе*.

### Структурная оптимизация ТП

Параметры структурной оптимизации:

- модели станков,
- типы инструментов,
- схемы базирования, т.е. варианты типовых решений.

Структурная оптимизация рассматривает последовательно каждую задачу технологического проектирования.

Весь процесс проектирования расчленяется на несколько взаимосвязанных уровней.

Процесс проектирования *на каждом уровне* представляет собой *многовариантную процедуру*.

В результате проектирования на всех уровнях образуется *граф допустимых вариантов ТП*, отвечающих заданным ограничениям.

**Задача структурной оптимизации** состоит в поиске ветви графа, обеспечивающей экстремум целевой функции.

В силу неупорядоченности параметров основной метод структурной оптимизации состоит в **последовательном переборе возможных вариантов**.

Чтобы *выбрать один оптимальный вариант*, необходимо до конца спроектировать очень большое количество допустимых техническими и технологическими ограничениями вариантов ТП.

Для реального ТП изготовления деталей даже средней сложности таких вариантов может быть *огромное множество*. Перебор всех вариантов даже при помощи современных быстродействующих компьютеров занимает *очень большое время*.

### Способы уменьшения времени проектирования:

**Прием 1.** Необходимо организовать *отбор рациональных вариантов проектных решений на каждом уровне проектирования*.

Однако при этом возникает проблема формирования *критериев промежуточного отбора наиболее рациональных вариантов на различных уровнях*.

Целесообразно, поэтому, использовать *в качестве критерия суммарную стоимость заготовки и механической обработки*. Однако стоимость механической обработки можно рассчитать только после разработки всего ТП. Следовательно, пропадает смысл «поэтапной оптимизации».

Но если удачно назначить критерии на каждом уровне проектирования, такой подход имеет смысл. При его применении может оказаться несколько равнозначных вариантов ТП, но среди них уже гораздо легче выбрать оптималь-

ный вариант.

**Прием 2. «Предпроектная оптимизация».**

**Прием 3.** Следующим шагом в развитии предпроектной оптимизации является *переход от булевых матриц соответствий к оценочным матрицам.*

Алгоритм поиска оптимального решения по оценочной матрице состоит в поиске одноименной строки в оценочных матрицах для всех диапазонов условий применимости, обеспечивающей наименьшую сумму затрат для данного условия задачи.

Рассмотренная процедура *повторяется* для каждого уровня проектирования, приводя в конечном итоге к варианту с оптимальной структурой.

**Параметрическая оптимизация ТП**

Заключается в расчете оптимальных припусков и межпереходных размеров, режимов резания и т.д.

## 7 СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ МАРШРУТНОЙ ТЕХНОЛОГИИ ТЕХНОПРО

### 7.1 Общие сведения о системе ТехноПро

Система *ТехноПро* предназначена для ручного, диалогового полуавтоматического и автоматического проектирования операционной технологии, включая операции: заготовительные, механической и термической обработки, нанесения покрытий, слесарные, технического контроля, сборки и другие (рисунок 7.1).



Рисунок 7.1 – Укрупненная схема комплекса

Система выдает в технологические процессы:

- наименования операций,
- оборудование,
- приспособления,
- вспомогательные материалы,
- формирует тексты переходов,
- рассчитывает технологические размеры с учетом припусков на обработку,
- обеспечивает подбор режущего, измерительного и вспомогательного инструментов.

Основным принципом работы *ТехноПро* является накопление знаний опытных технологов конкретного предприятия с последующим использованием этого опыта независимо от них. То есть система позволяет аккумулировать опыт наиболее квалифицированных специалистов предприятия, использовать и

тиражировать его, обучать на его основе молодых специалистов. Это свойство системы особенно важно в настоящее время, когда опыт проектирования технологии, накопленный за десятилетия работы, уходит с предприятий вместе с технологами пенсионного возраста.

В *ТехноПро* заложена возможность ее обучения пользователями и самообучения системы в процессе работы. Обучение системы ведется на основе технологических понятий без какого-либо формализованного языка программирования. Интерфейс *ТехноПро* с пользователем реализован в стиле *Microsoft Office*.

Исходной информацией для обучения системы являются ТП изготовления конкретных изделий, которые уже отлажены на производстве. По мере наполнения баз данных система обретает возможность проектирования технологии изготовления совершенно новых изделий, которых еще не было в производстве.

На рисунке 7.2. представлены основные виды информации, которыми пользователь может оперировать при диалоговом проектировании ТП.



Рисунок 7.2 – Информация, используемая пользователем при диалоговом проектировании технологических процессов

Наряду с оригинальным методом проектирования по “общим технологическим процессам” в *ТехноПро* реализованы и традиционные методы: по типовому, групповому, технологическому процессу-аналогу. Технолог сам выбирает метод проектирования, наиболее подходящий в конкретном случае, а также способ его использования: автоматический, полуавтоматический, диалоговый или их сочетание. Например, сборочные технологические процессы можно проектировать в диалоге, изготовление корпусных деталей - в полуавтоматическом режиме, а процессы изготовления тел вращения - в автоматическом.

В отличие от других систем *ТехноПро* позволяет опытному технологу один раз внести правила выбора того или иного маршрута, операции, перехода, инструмента или другого компонента ТП и далее система будет использовать их автоматически. При этом технолог полностью уверен, что система спроектирует ТП, точно соответствующий его опыту. Время автоматического проектирования ТП составляет секунды, что освобождает специалистов от длительного

формирования ТП и заполнения карт. Менее опытные специалисты могут ограничиться освоением только "проектной" части системы, не изучая средства формирования баз данных, и использовать в своей работе заложенные опытными технологами решения и правила.

*ТехноПро* формирует операционные, маршрутно-операционные и маршрутные технологические карты, карты контроля, ведомости оснастки или материалов, титульные листы и другие технологические документы. На многих предприятиях используемые технологические карты отличаются от карт, принятых по ГОСТ. *ТехноПро* обеспечивает автоматическое заполнение технологических документов произвольных форм, созданных в текстовом редакторе *Microsoft Word*.

Система *ТехноПро* может использоваться автономно (с вводом информации о конструкции деталей с чертежей на бумаге), либо совместно с системами автоматизированного конструирования.

В этом комплексе чертежи, выполненные в системе T-FLEX CAD, поступают через интерфейс OLE Automation в *ТехноПро* для проектирования технологии. По спроектированным технологическим процессам могут быть сформированы заказы на конструирование оснастки, технологию изготовления которой также можно разработать в *ТехноПро*. Получаемая конструкторская и технологическая документация хранится в системе электронного документооборота T-FLEX DOCs.

Важным свойством системы является её автоматическая реакция на изменение в конструкции детали или в требованиях к качеству поверхностей. В зависимости от требования к качеству изготовления поверхностей детали система автоматически подбирает необходимые операции и рассчитывает технологические размерные цепи.

## **7.2 Методика автоматизированного проектирования ТП в САПР ТехноПро**

### ***А) Предварительный этап***

***Необходимо создать базу данных.*** Для этого нужно сгруппировать детали, в основном по сходству технологии их изготовления. При этом для каждой группы создается общий технологический процесс, который содержит весь перечень операций изготовления всех деталей группы. Для создания общего технологического процесса используются технологические процессы, уже освоенные в производстве. Можно использовать «бумажные» варианты технологических процессов с последующим их «превращением» в электронный вариант или конкретные технологические процессы, созданные в ходе работы с *ТехноПро* в диалоговом режиме.

### ***Б) Создание общего технологического процесса***

1) Один из технологических процессов группы принимается за базовый и

вводится в виде общего технологического процесса (можно скопировать один из конкретных технологических процессов, созданных в ходе работы в *ТехноПро* в диалоговом режиме).

2) В него добавляются недостающие операции и переходы из других технологических процессов (конкретных технологических процессов).

При добавлении выявляются признаки, в зависимости от которых необходимо выбирать ту или иную операцию, переход или маршрут. Проверка каждого из признаков вносится в виде условий в базу ТехноПро.

Примерами таких условий являются проверки: вида заготовки, марки или твердости материала детали, габаритов детали, наличия определенных элементов конструкции (поверхностей), их размеров и т.д.

### ***В) Автоматическое проектирование индивидуальных технологических процессов***

3) создается описание конструкции детали, для которой необходимо спроектировать технологический процесс ее обработки.

Описание можно создать:

– (**считать автоматически**) с электронной версии параметрического чертежа детали, созданного посредством системы геометрического моделирования **T – FLEX** (разработка уже упомянутой ранее фирмы «Топ Системы»).

– (**интерактивно**) вводя необходимые данные с клавиатуры.

– (**по шаблону**) можно скопировать подобную деталь из уже имеющихся в базе системы **конкретных технологических процессов** или скопировать макет общего технологического процесса.

**Описание чертежа детали заключается в заполнении:**

– общих сведений о детали (**данные из штампа и технические требования чертежа**) и

– параметров элементов конструкции (поверхностей), имеющих на чертеже детали.

4) Детали назначается общий технологический процесс соответствующей группы деталей и запускается процесс автоматического формирования технологического процесса.

В течение этого процесса ТехноПро

– выбирает из назначенного общего технологического процесса операции и переходы, необходимые для изготовления каждого элемента конструкции детали и переносит их в конкретный технологический процесс.

– Затем из выбранного перечня система отбрасывает операции и переходы, обеспечивающие лучшее качество изготовления поверхностей детали по сравнению с указанными требованиями на чертеже.

5) Система отбрасывает из конкретного технологического процесса операции и переходы, в которых условия их выбора не выполнены.

6) ТехноПро производит расчеты, имеющиеся в условиях оставшихся операций и переходов.

7) Система рассчитывает технологические размерные цепи с учетом значе-

ний припусков, указанных в переходах общего технологического процесса.

8) Система выполняет условия подбора технологического оснащения операций и переходов и выполняет имеющиеся в этих условиях расчеты режимов обработки и норм изготовления.

9) Система формирует тексты переходов, заменяя имеющиеся в них параметры на конкретные значения. Значения параметров выбираются в зависимости от типа выполняемой обработки – предварительной или окончательной.

Создавая общие технологические процессы и условия, технолог «обучает» ТехноПро проектированию технологии конкретного (своего) производства. Однажды обучив систему, технолог может быть уверен, что ТехноПро никогда не забудет производственных нюансов проектирования технологических процессов.

### **7.3 Алгоритм проектирования ТП в САПР ТехноПро**

Система *ТехноПро* позволяет проектировать как технологию изготовления механообрабатываемых деталей, так и другие виды технологий. Наполнение баз системы определяет вид проектируемой технологии, например, нанесения покрытий, термообработки, штамповки, сварки, сборки, электромонтажа, изготовления печатных плат и других.

При проектировании техпроцесса в диалоговом режиме наименование операций, нормировочные данные, составление переходов и т.д. производится путем ввода информации с клавиатуры.

База данных хранения технологических операций, переходов и т.д. разбита на две составляющие: общие технологические процессы (ОТП) и конкретные технологические процессы (КТП).

При проектировании ТП технолог оперирует знакомыми ему понятиями – Спецификация, Деталь, Операции, Переходы, Оснащение, Таблицы, Карты, Эскизы, Комментарии. Заполняя сведения о детали, можно вводить с клавиатуры или считывать с электронного чертежа: наименование, обозначение детали, материал, профиль и размеры заготовки, другие сведения (рисунок 7.3). Формирование сборочного ТП производится на основе спецификации (рисунок 7.4).

Представление маршрута операций и переходов в виде «дерева» существенно упрощает формирование ТП. При указании курсором на операцию или переход справа открывается форма для их заполнения (рисунок 7.5, 7.6). Порядок следования операций или переходов можно изменить нажатием кнопок со стрелками вверх или вниз, при этом номера операций или переходов пересчитываются автоматически.

В «ТехноПро» проектирование ТП может проводиться в различных режимах. Технолог сам выбирает оптимальный режим или их сочетание. Например, первоначальное наполнение баз системы производится в режиме диалога, затем можно перейти к проектированию с использованием техпроцесса-аналога.

В базе данных Конкретных технологических процессов (КТП) для каждого

технологического процесса имеются поля ссылок на чертежи или модели деталей. Используя средства классификации и поиска, просматривая чертеж или трехмерную модель, технолог находит требуемый техпроцесс-аналог.

Найденный техпроцесс-аналог копируется с новым именем и обозначением, а затем просматривается и корректируется. Корректировка заключается в добавлении, удалении, редактировании или изменении положения в маршруте операций и переходов.

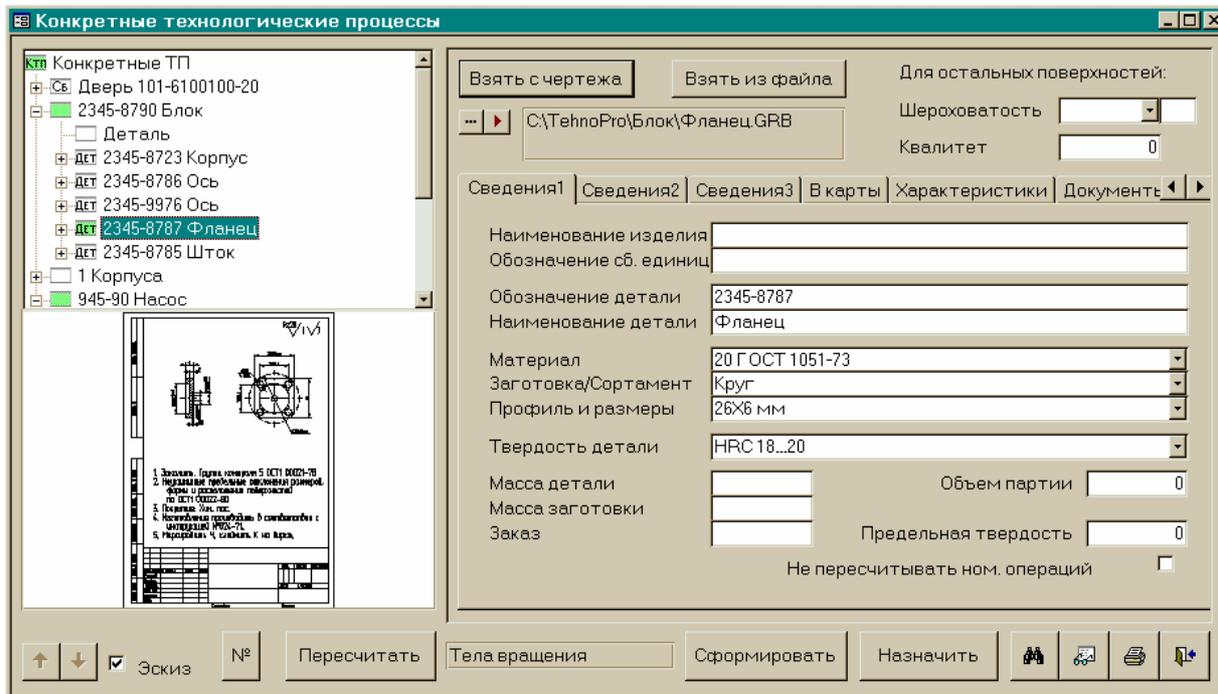


Рисунок 7.3 – Общие сведения о детали

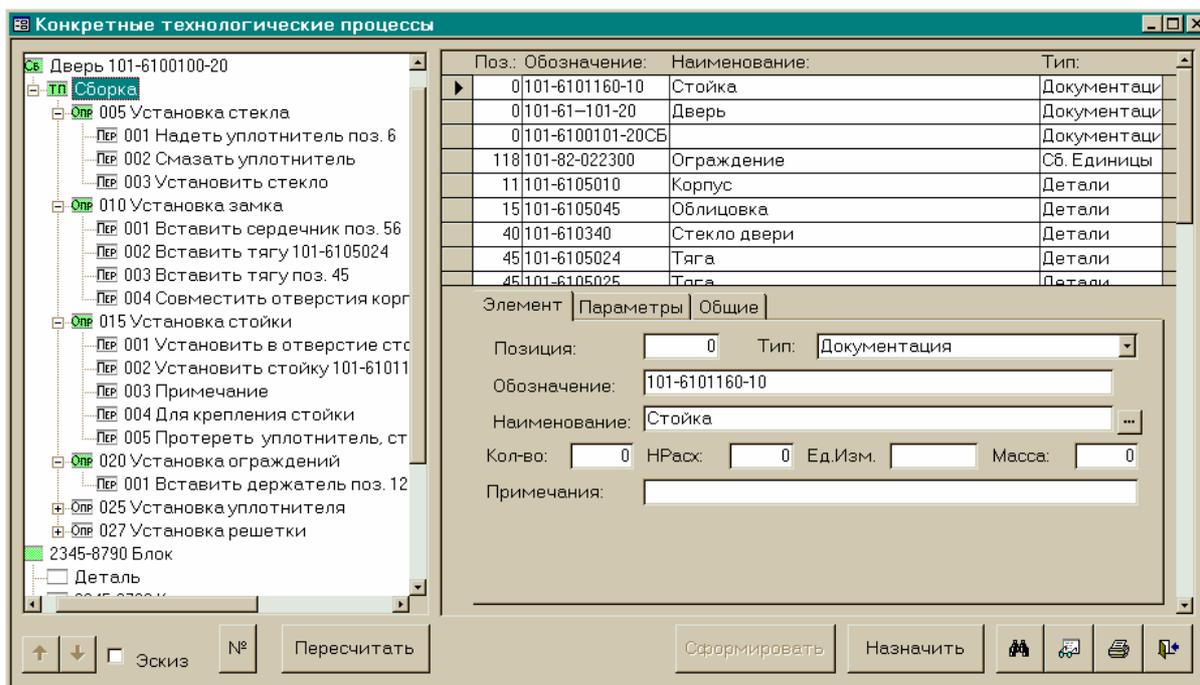


Рисунок 7.4 – Спецификация

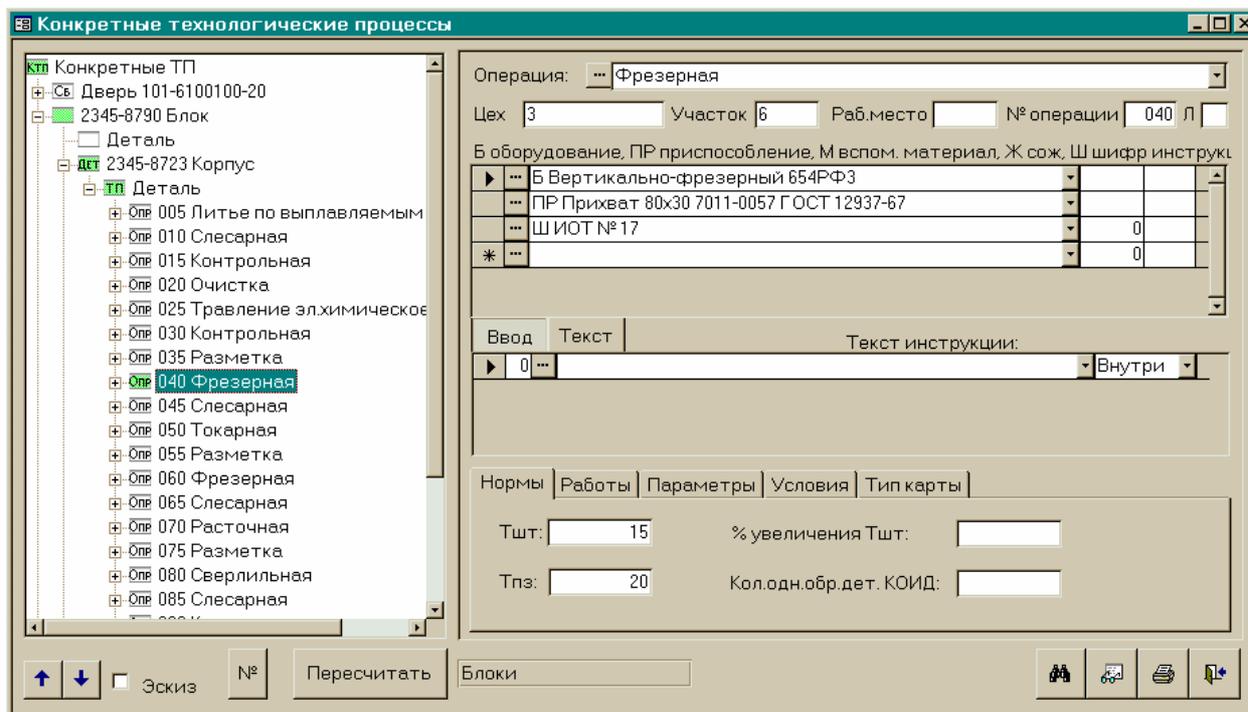


Рисунок 7.5 – Заполнение содержания операции

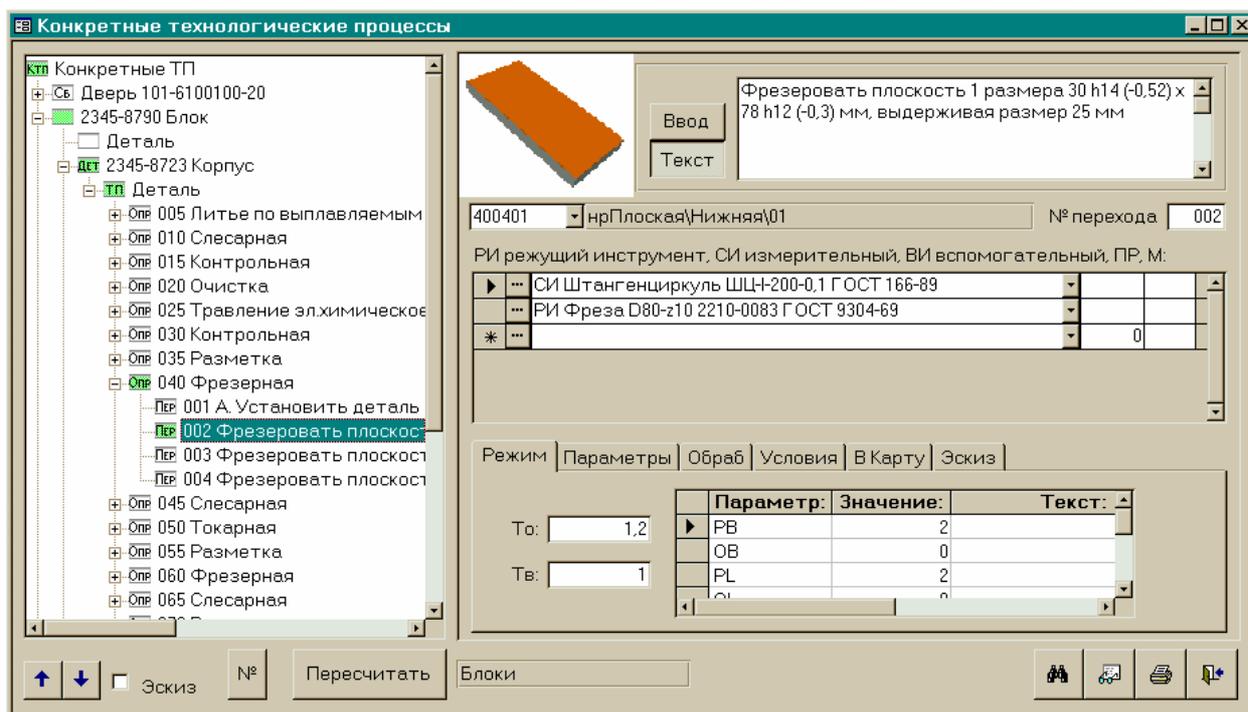


Рисунок 7.6 – Заполнение содержания перехода

При корректировке ТП выбор оснащения (оборудования, приспособлений, инструментов, материалов) производится по технологическому классификатору из Информационной базы системы. Записи оснащения могут иметь различную структуру, которая легко может быть перенастроена пользователем. Каждая запись может содержать параметры для автоматического подбора или поиска в

диалоге, а также может быть снабжена иллюстрацией (рисунок 7.7). Каждое оборудование снабжается паспортными данными.

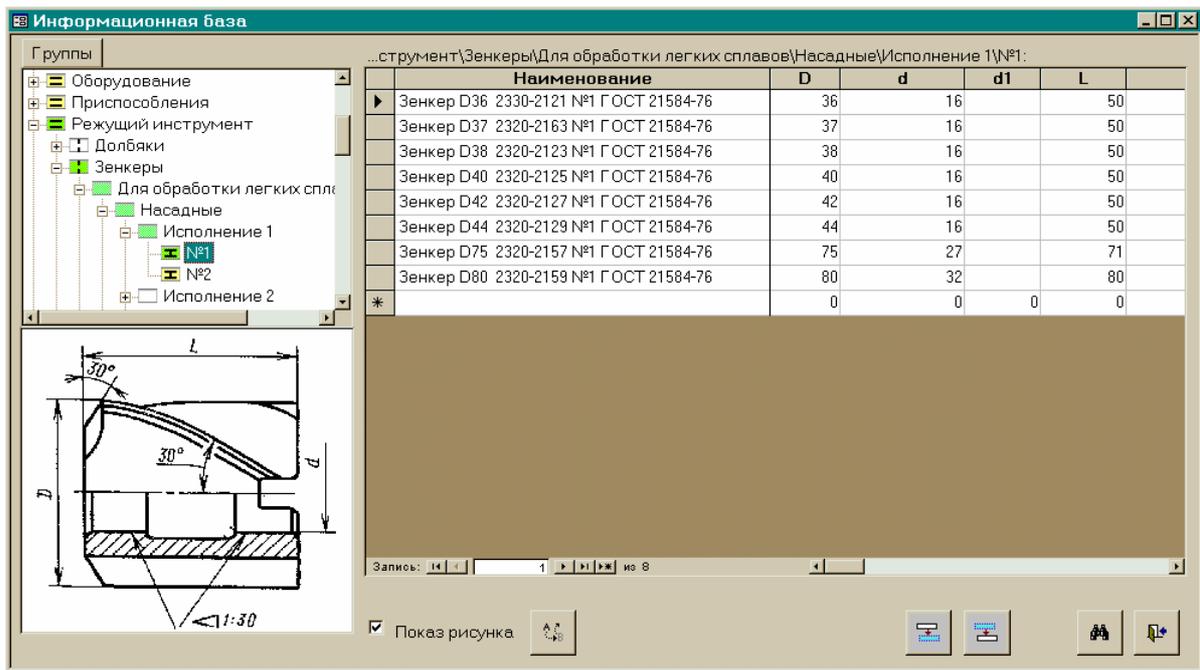


Рисунок 7.7 – Выбор оснащения

«ТехноПро» обеспечивает гибкость настройки выбора данных из справочных баз, таблиц и проведение расчетов. Пользователь может сформировать «Сценарии», состоящие из набора процедур и подключить их к любому полю операции, перехода, оснащения или группам Информационной базы. При этом нет необходимости заполнения никаких промежуточных таблиц – достаточно указать связи курсором (рисунок 7.8).

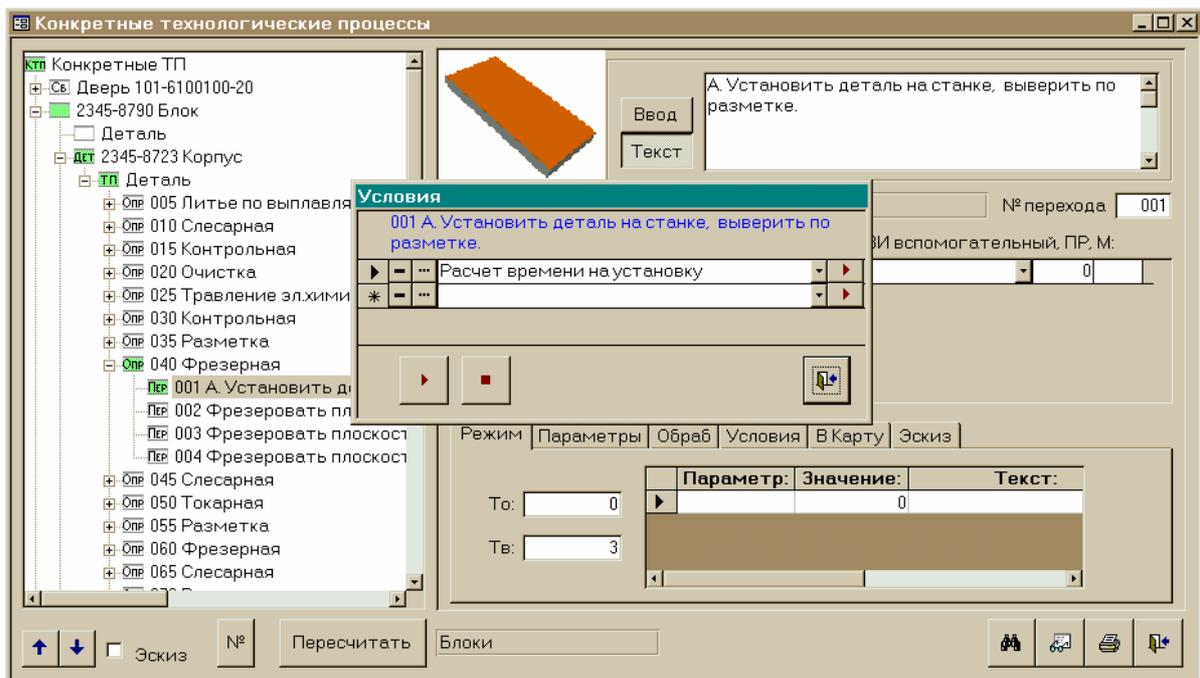


Рисунок 7.8 – Создание сценария

Такой подход позволяет определять последовательность проектирования, при которой предлагаемые технологу варианты зависят от выбора, выполненного на предыдущем шаге проектирования. Например, в зависимости от наименования операции, пользователю предлагается список оборудования, на котором возможно ее выполнение. Далее выбираются цех и профессия работника. В операцию автоматически вносятся номер инструкции по охране труда, необходимые приспособления и прочее.

Технологические операционные эскизы и карты наладки могут выполняться в любом графическом редакторе. В разрабатываемом ТП эти графические документы подключаются к переходам операций с возможностью их просмотра (рисунок 7.9). При необходимости пользователь может отредактировать изображение, а также просмотреть его в увеличенном виде. Изображение эскизов и карт наладки автоматически вставляются в формируемые технологические карты.

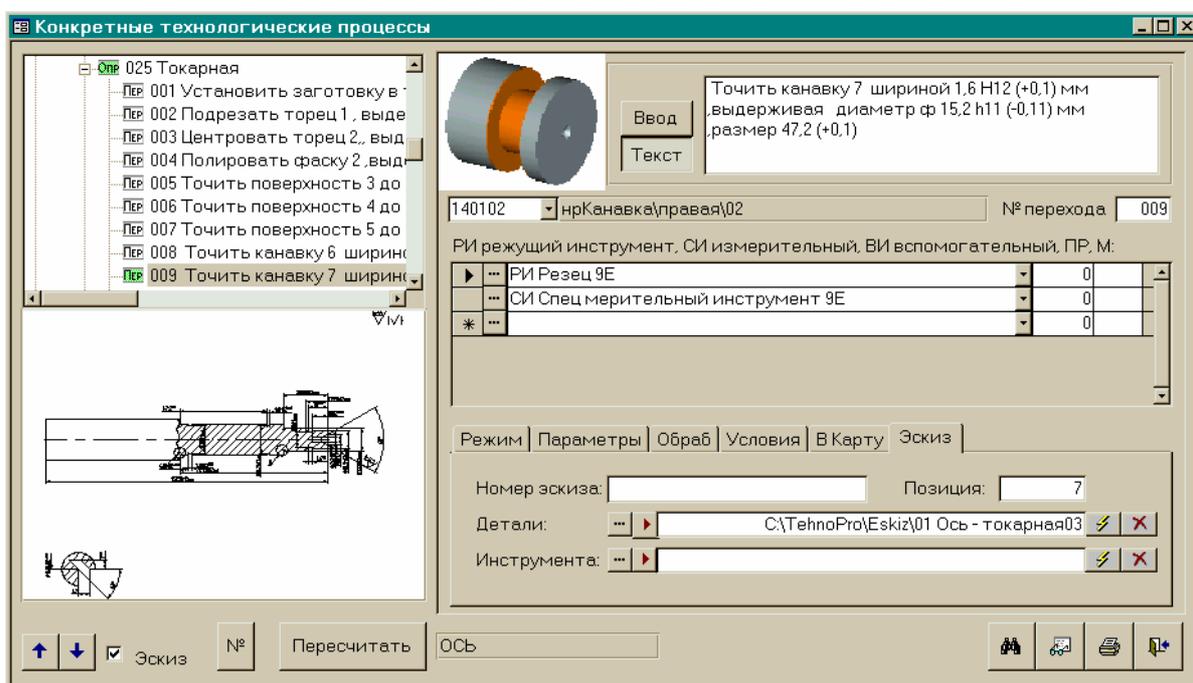


Рисунок 7.9 – Подключение и просмотр эскизов

Для ускорения формирования ТП пользователь может создавать наборы операций или типовых, групповых, единичных технологических процессов. Можно использовать принцип дублирования технологий (рисунок 7.10).

При использовании на предприятии типовых или групповых технологических процессов «ТехноПро» обеспечивает возможность их параметризации. Такие параметрические ТП могут автоматически пересчитываться. Причем информация для пересчета (описание конструкции) может быть получена из конструкторских САПР или введена вручную с чертежа на бумаге.

«ТехноПро» органично включается в интегрированные комплексы с любыми современными конструкторскими САПР, так как может поставляться с интерфейсами к множеству систем. Например, SolidWorks, Компас-График, Pro/ENGINEER и другими. Опыт эксплуатации системы «ТехноПро» на сотнях

предприятий также показал высокую степень готовности базы данных этой системы к интеграции с многими АСУП и системами PDM.

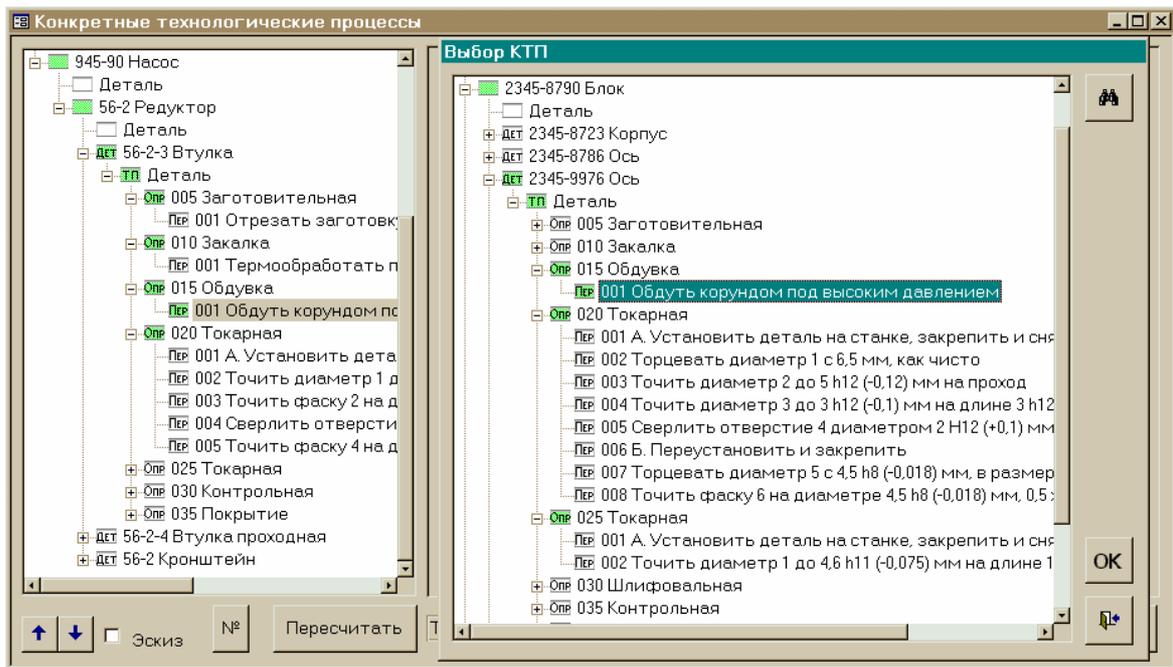


Рисунок 7.10 – Дублирование технологии

Входными данными для систем управления предприятиями являются трудовые и материальные нормативы. Для их расчета в «ТехноПро» реализованы методы укрупненного табличного и точного аналитического расчетов, а также возможность их сочетания. Система позволяет пользователю ввести любые таблицы с данными (рисунок 7.11), расчетные формулы (рисунок 7.12) и правила их взаимодействия. При этом участия программистов не требуется.

	Lmin	Lmax	Xar	Xar_vid	Dmin	Dmax	Prip_L
...	0	3	подрезание	предварительное	18	50	0,6
...	0	3	подрезание	чистовое	18	50	0,4
...	0	3	шлифование		18	50	0,2
...	3	6	подрезание	предварительное	0	3	0,28
...	3	6	подрезание	чистовое	0	3	0,18
...	3	6	шлифование		0	3	0,1
...	3	6	подрезание	предварительное	3	6	0,32
...	...	...	...	...	6	6	0,2
...	...	...	...	...	6	6	0,15
...	...	...	...	...	6	10	0,36
...	...	...	...	...	10	10	0,26
...	...	...	...	...	10	10	0,15
...	...	...	...	...	18	18	0,5
...	...	...	...	...	18	18	0,3
...	...	...	...	...	18	50	0,2
...	...	...	...	...	50	50	0,6
...	...	...	...	...	50	50	0,4
...	...	...	...	...	50	10	0,2
...	...	...	...	...	3	3	0,32
...	...	...	...	...	3	3	0,22
...	...	...	...	...	3	3	0,1
...	...	...	...	...	6	6	0,36
...	...	...	...	...	6	6	0,24
...	6	10	шлифование		3	6	0,15
...	6	10	подрезание	предварительное	6	10	0,4
...	6	10	подрезание	чистовое	6	10	0,3

Рисунок 7.11 – Данные и иллюстрации в таблицах

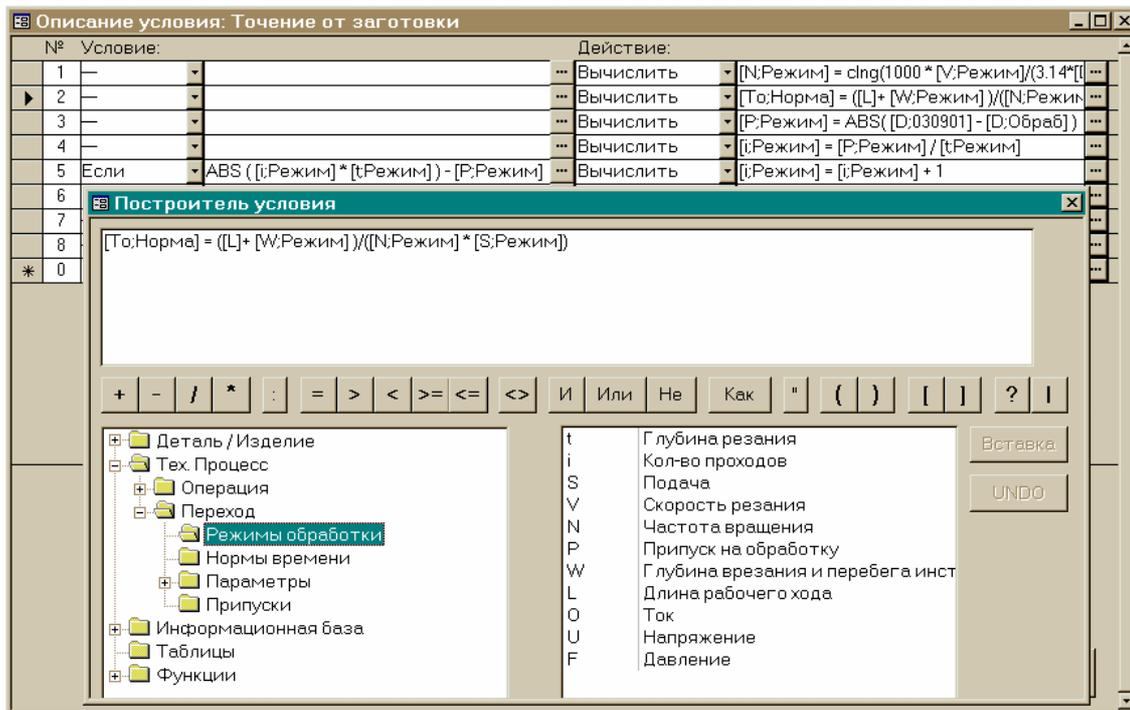


Рисунок 7.12 – Формулы, логика и их построитель

«ТехноПро» поставляется с множеством форм выходных документов. Формы выполнены в MS Word (рисунок 7.13). Поэтому новые формы (горизонтальные и вертикальные) создаются пользователями без участия программистов. Поддерживается использование специальных символов: угол, градус и другие. В поставку системы входят тысячи информационных групп данных более чем по 300 ГОСТ. Состав и структура баз может расширяться пользователями самостоятельно.

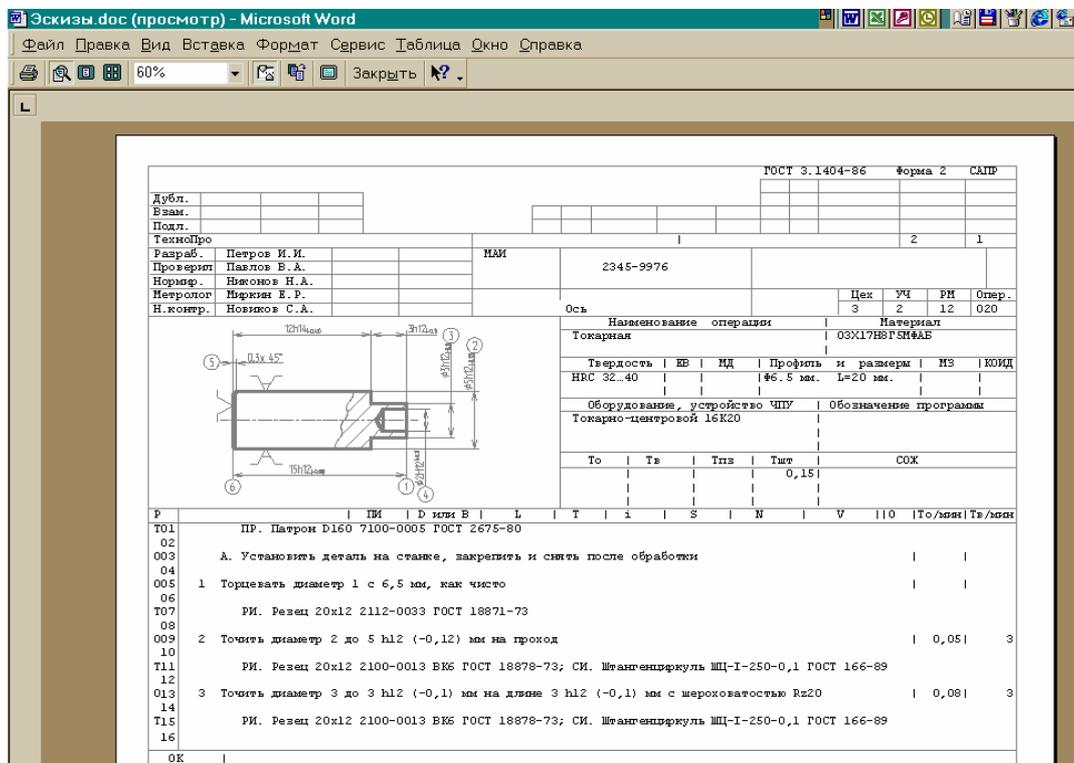


Рисунок 7.13 – Пример сформированной карты ТП

При разработке нового ОТП в ветви «Наборы операций» добавляется новый пункт, например «Технология РЭС» (рисунок 12.14), в макете технологического процесса последовательно добавляются технологические операции.

Если в информационной базе имеется аналогичная операция, программа автоматически найдет ее и предложит пользователю ее использовать с уже выбранным оборудованием, оснасткой, технологическими переходами. Если аналогичной операции в информационной базе нет, то в поле «Операция» вводится наименование операции, ставится номер цеха, участка и рабочего места. Затем производится выбор оборудования из информационной базы.

Далее определяется наличие технологических переходов в операции и производится быстрый поиск соответствующего перехода в информационной базе. Если необходимого технологического перехода в информационной базе нет, его предварительно вводят в нее (рисунок 7.15) и затем выбирают операцию.

Все эти этапы повторяются до тех пор, пока все технологические операции не будут сформированы полностью.

При разработке КТП вначале ищутся все технологические операции в ОТП, которые копируются в КТП (рисунок 7.16). После этого формируются все остальные операции и технологические переходы (рисунок 7.17) аналогично ОТП. Нумерация операций и переходов производится автоматически.

После просмотра и, при необходимости, редактирования спроектированного ТП, его можно выдать на печать, выбрав необходимый шаблон маршрутно-операционной карты (рисунок 7.18, 7.19).

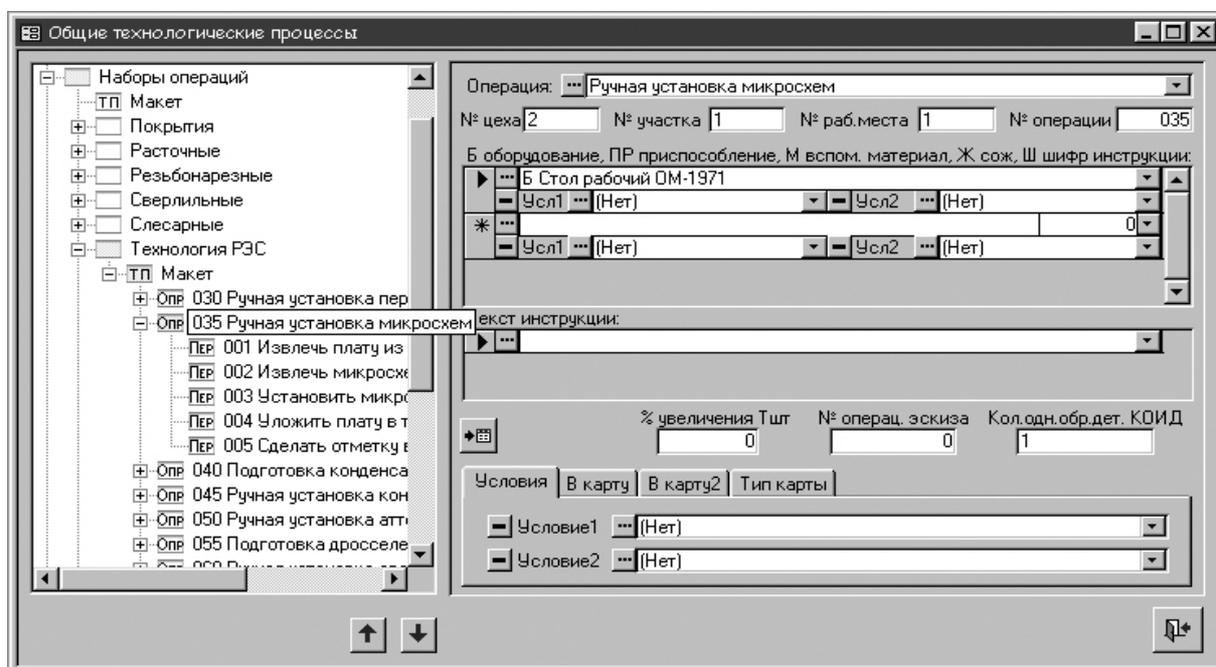


Рисунок 7.14 – Проектирование ОТП. Формирование технологической операции

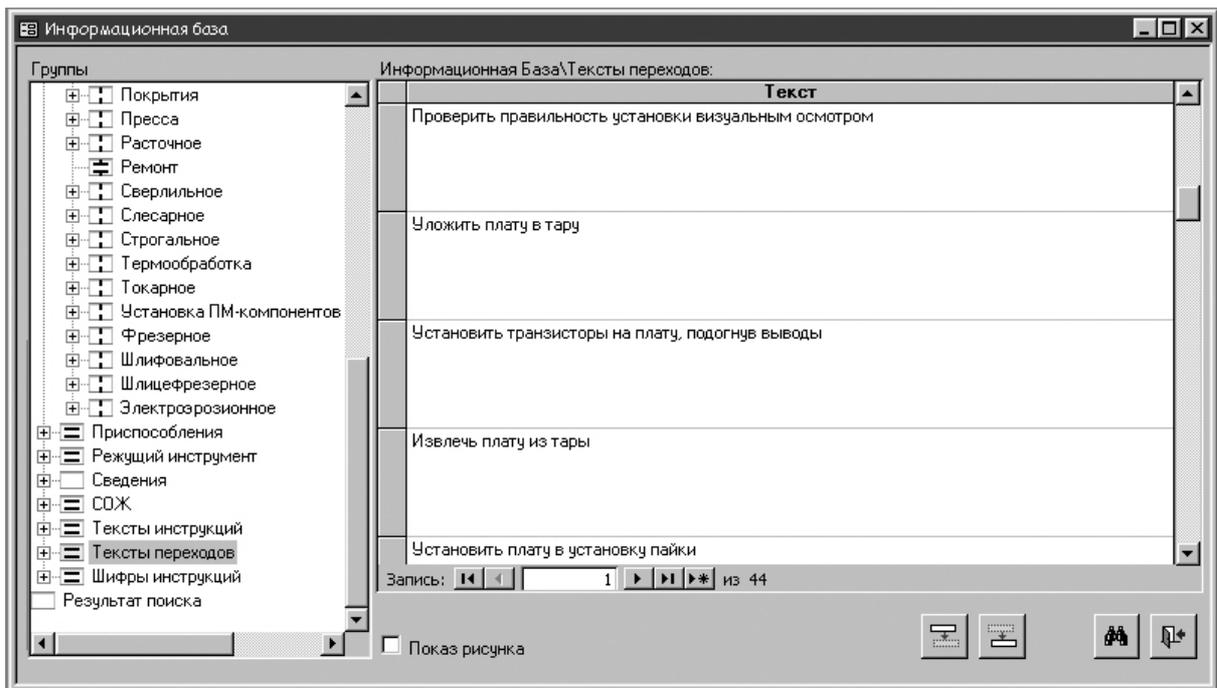


Рисунок 7.15 – Ввод нового технологического перехода

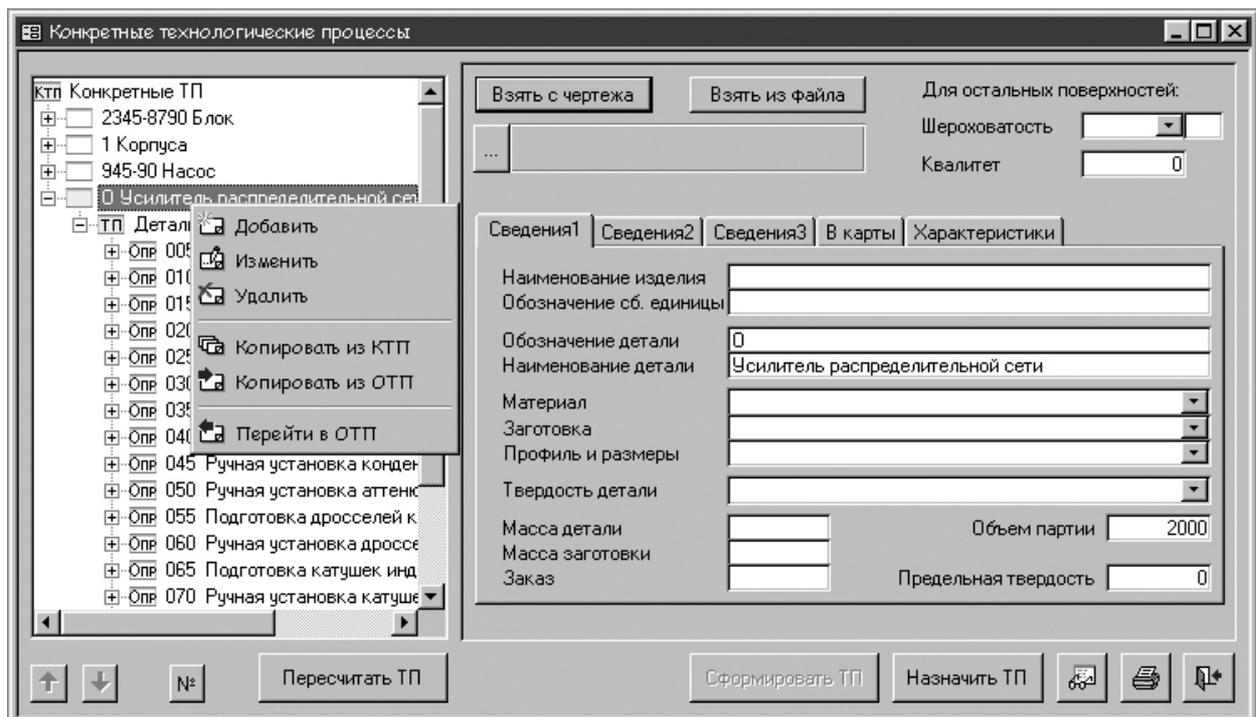


Рисунок 7.16 – Вид диалогового окна копирования операций из ОТП

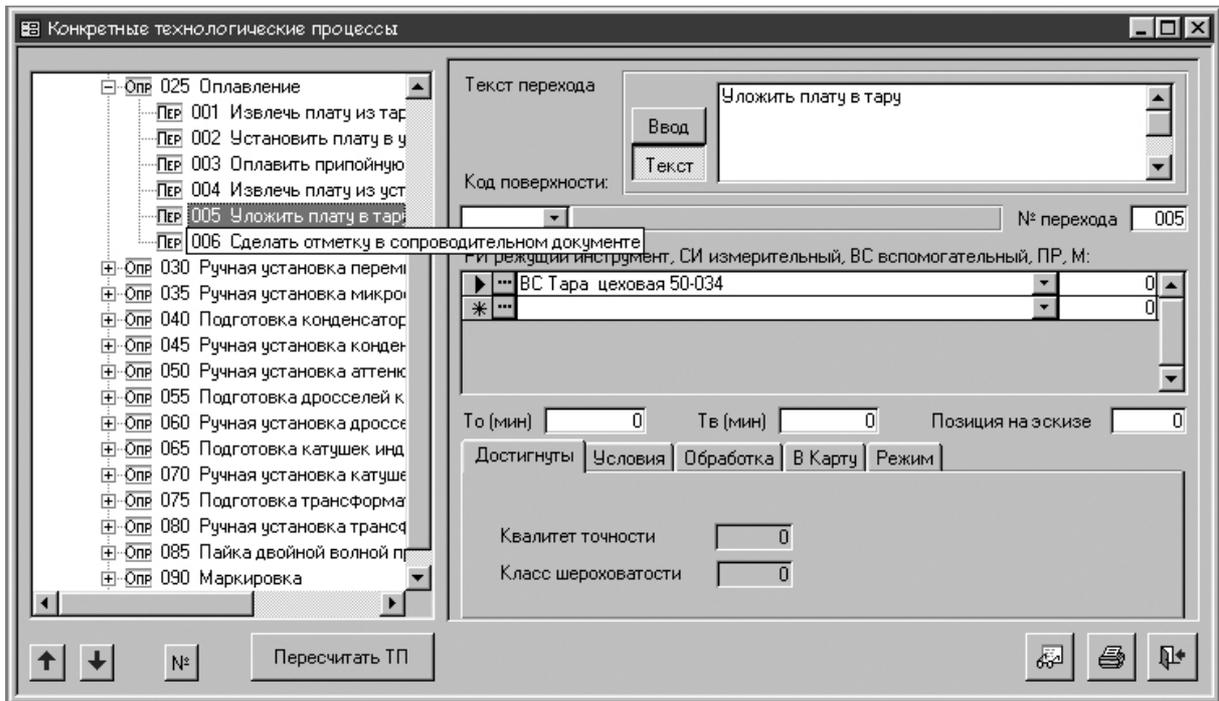


Рисунок 7.17 – Ввод технологического перехода при проектировании КТП



Рисунок 7.18 – Вид панели при выводе технологических документов на печать

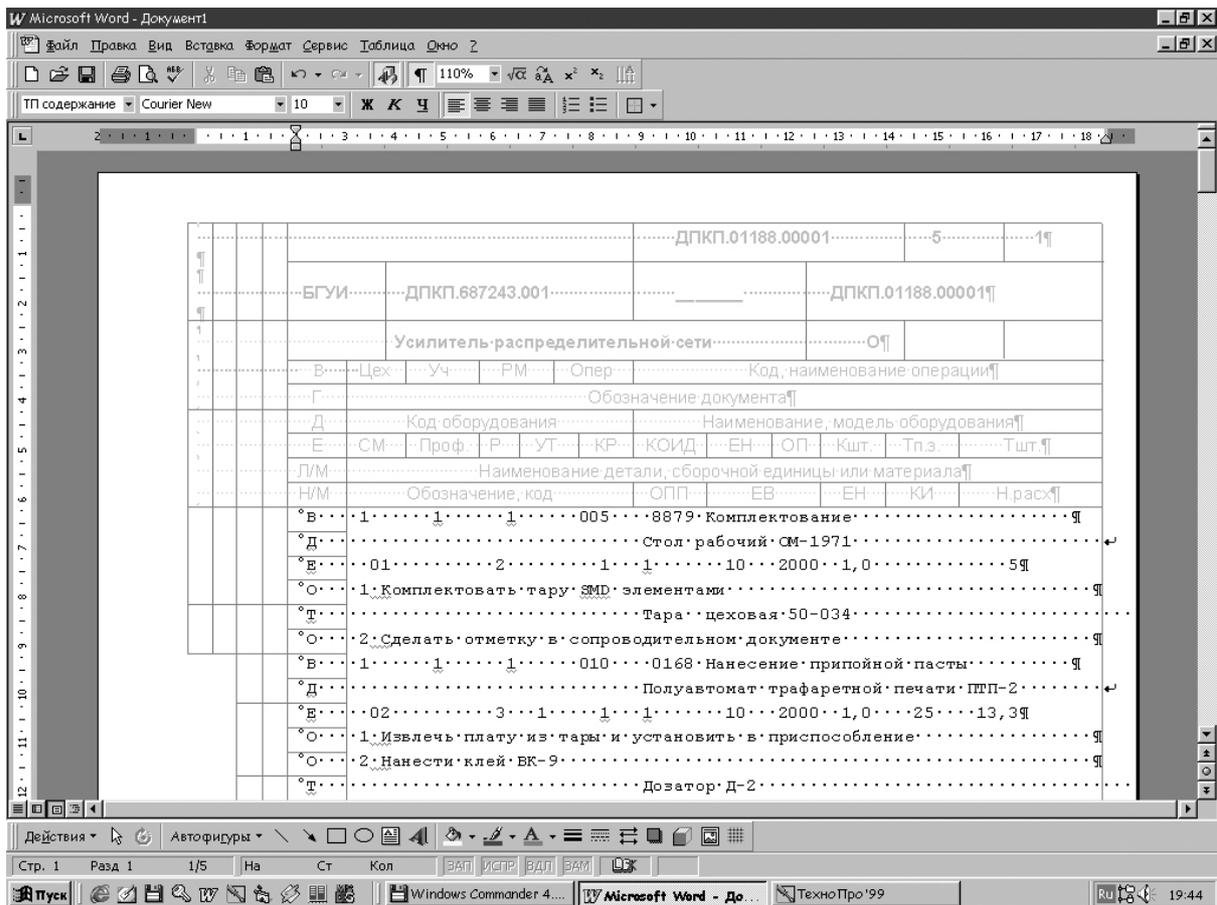


Рисунок 7.19 – Сформированная маршрутно-операционная карта

В основе принципа автоматического проектирования технологического процесса лежит подготовка условий для выбора режимов, оборудования, материалов и т.д.. Для автоматического проектирования необходимо подготавливать технологические операции в ОТП, в которые затем включаются разработанные условия и уже при разработке КТП выбираются предварительно разработанные операции из ОТП.

Разработка условий производится с помощью построителя условий в диалоговом режиме, в котором можно построить любое расчетное или логическое условие для определенного действия в момент проектирования технологического процесса.

Например, подбор оборудования для операции дозированного нанесения адгезива и условие для геометрических размеров печатной платы выглядит так:

Подобрать [Нанесение адгезива припойных паст;Vmax] <= [V]

И

[Нанесение адгезива припойных паст;Lmax] <= [L].

После того как все условия написаны и сохранены, производится их подключения в операции и переходы в ОТП (рисунок 7.20).

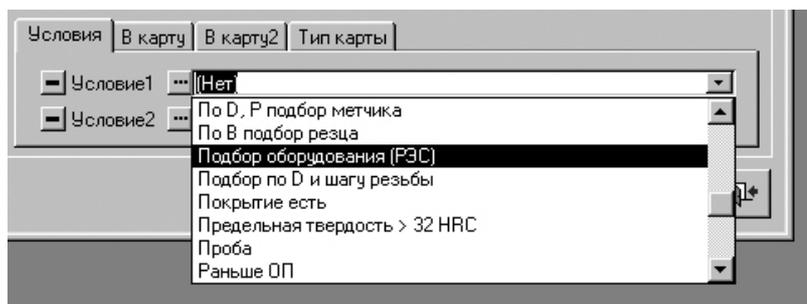


Рисунок 7.20 – Подключение условий в операциях и переходах

При формировании КТП производится выбор соответствующих операций и переходов из ОТП. После подготовки всего технологического процесса вводятся все необходимые параметры на разрабатываемую технологию изделия (рисунок 7.21). После нажатием кнопки «Пересчитать ТП» производится считывание условий, по которым рассчитываются все значения и выполняются соответствующие действия: подбор оборудования, оснастки режимов и т.д.

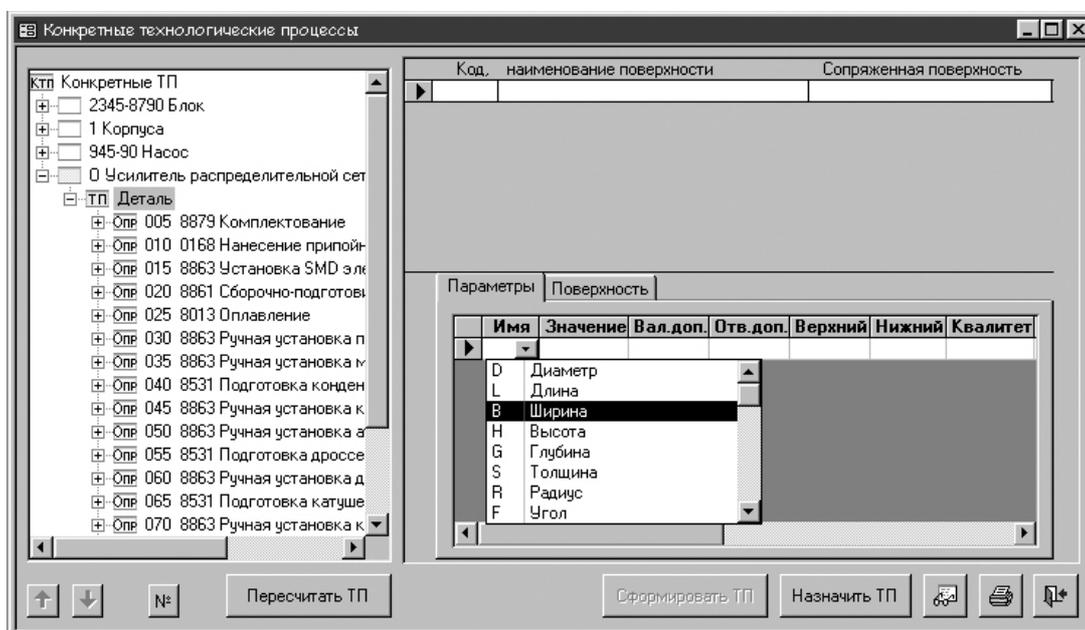


Рисунок 7.21 – Ввод параметров на окончательном этапе разработки технологии

Проверив правильность всех расчетов и действий можно выводить документ на печать.

## Инструкция по созданию конкретного технологического процесса в САПР ТехноПро

### 1 Вход в систему

По умолчанию войти в программу можно, набрав в соответствующем диалоговом окне в полях Имя и Пароль: SA (System Administrator) (рисунок 7.22)

### 2 Создание КТП

Для создания КТП в Основном меню выбираем пункт «Конкретные Тех. Процессы» (рисунок 7.23)

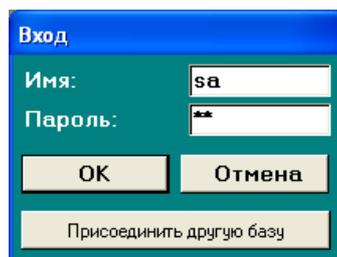


Рисунок 7.22 – Вход в программу

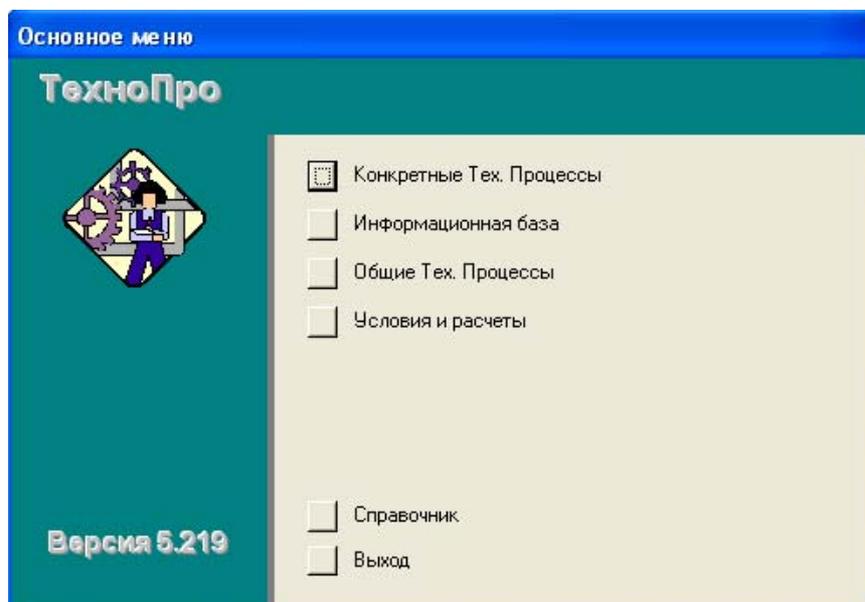


Рисунок 7.23 – Вход в конкретные технологические процессы

Затем в окне «Конкретные технологические процессы» правой клавишей мыши жмем на «Конкретные ТП» и выбираем «Добавить СБ» (рисунок 7.24).

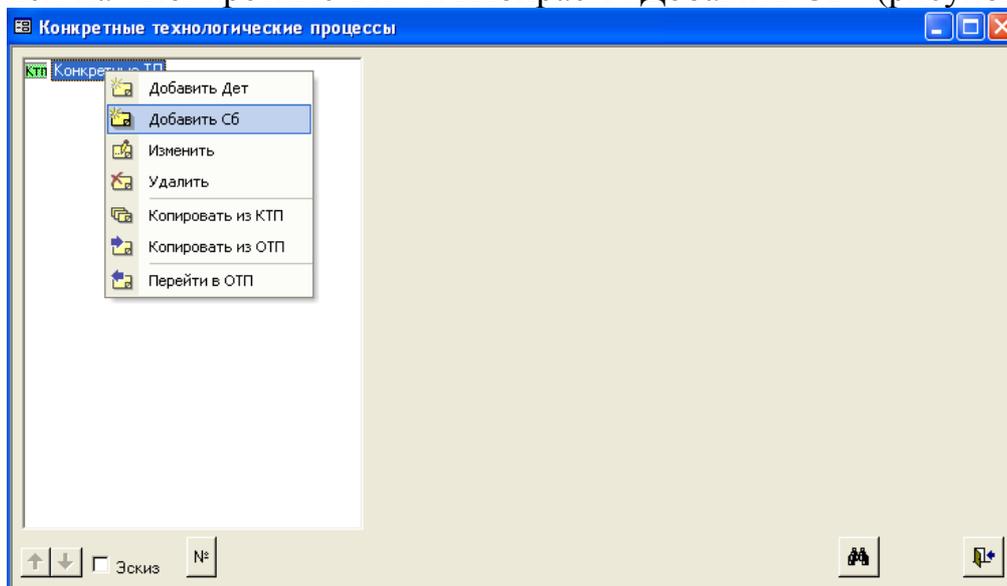


Рисунок 7.24 – Добавление техпроцесса для сборочного изделия

Далее в появившемся окне обязательно заполняем поля «Обозначение сборки» и «Наименование сборки» (рисунок 7.25).

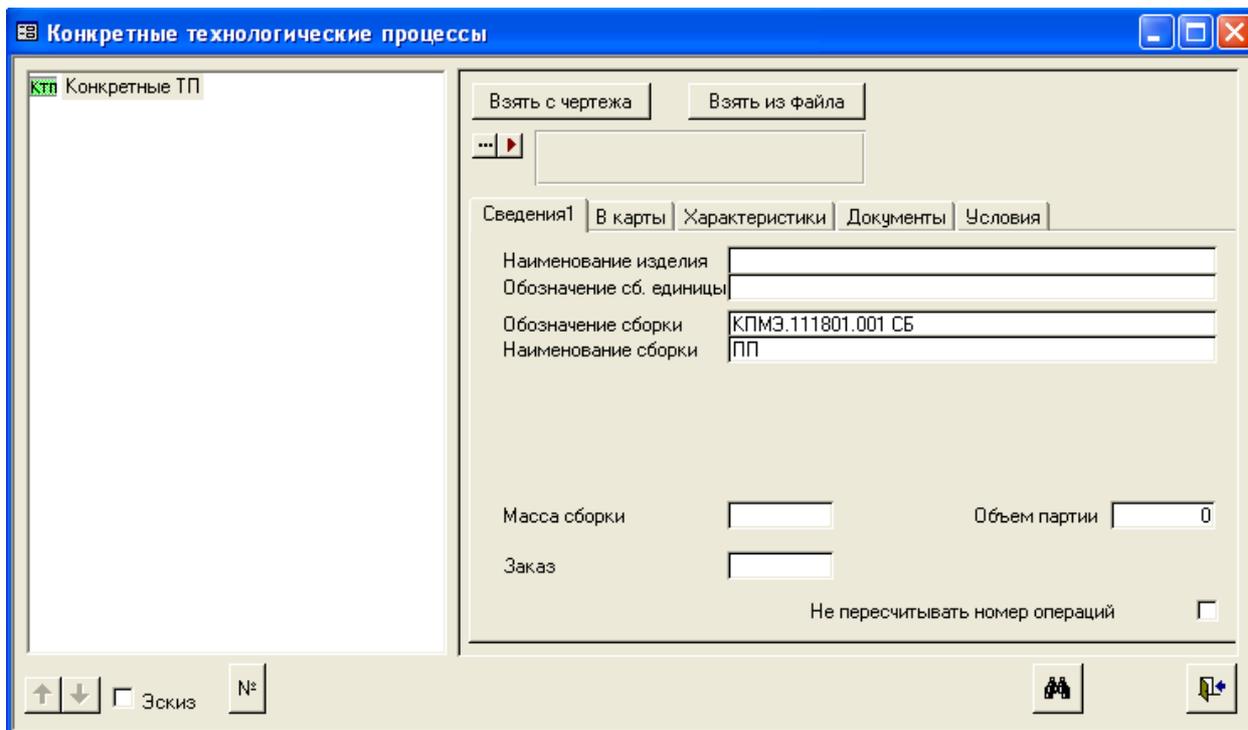


Рисунок 7.25 – Заполнение полей КТП.

Необходимо заполнить поля под закладкой «В карту». Следующим шагом нужно правой клавишей мыши кликнуть на «ТП Сборка» и выбрать из выпадающего меню пункт «Копировать из ОТП» (рисунок 7.26)

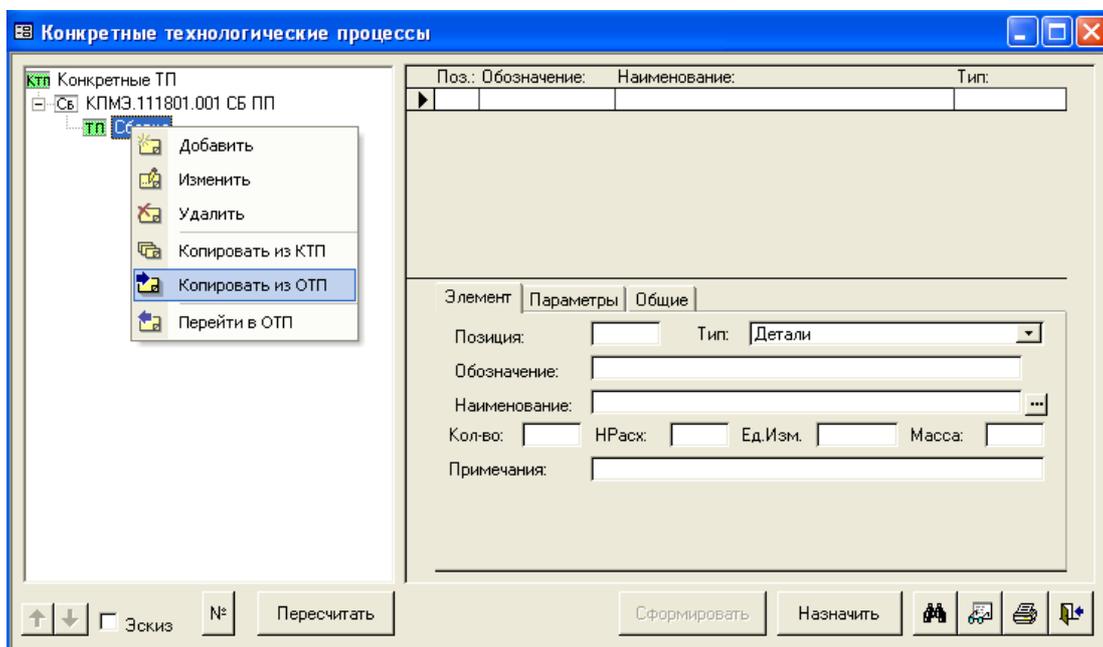


Рисунок 7.26 – Переход в «Общие тех. процессы»

В появившемся окне выбрать нужную операцию и нажать клавишу «ОК».

Выполнять это действие до тех пор, пока все нужные операции не будут добавлены в проектируемый КТП. По окончании добавления выйти из ОТП, нажав специальную клавишу выхода (рисунок 7.27).

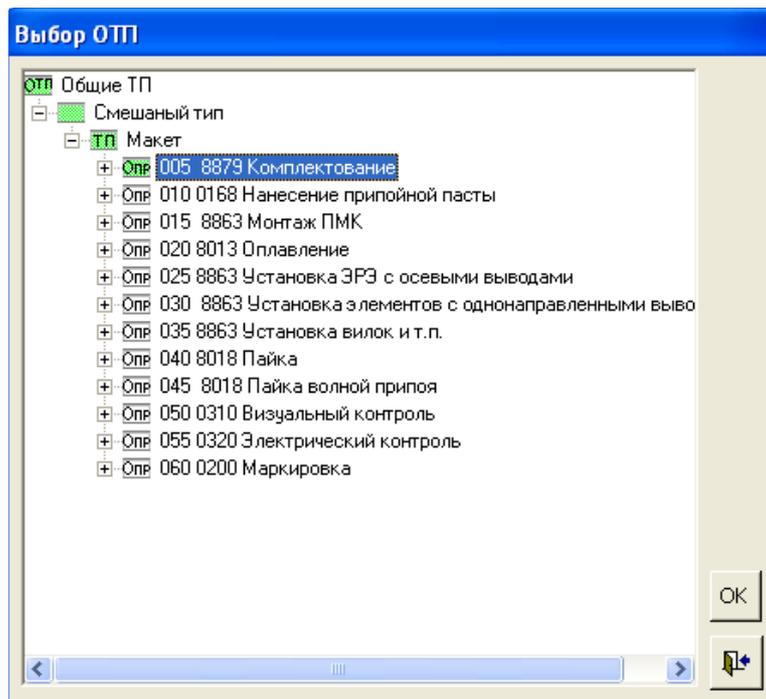


Рисунок 7.27 – Добавление операций из ОТП в проектируемый КТП

Потом в каждой операции редактируем необходимые для заполнения поля в закладках «Нормы», «Работы», а также полей «Цех», «Участок» и «Раб. место» (рисунок 7.28).

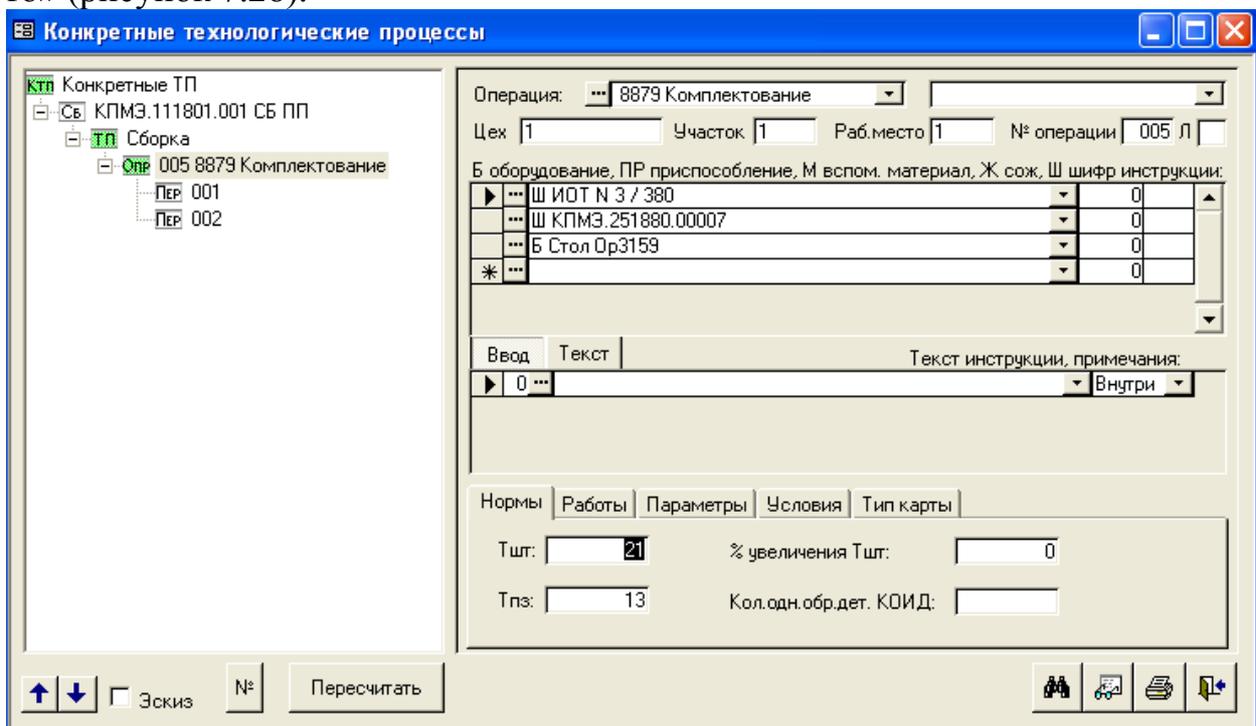


Рисунок 7.28 – Заполнение полей

Также редактируем Шифры инструкций (если это необходимо). Новые Шифры инструкций занесутся в Информационную базу программы.

Для генерации Маршрутно-операционной карты нужно нажать клавишу



, а затем клавишу  напротив «КТП ф1. Маршрутно-операционный ТП».

Маршрутно-операционная карта сформируется в программе MS Word в автоматическом режиме. После формирования карта будет доступна к редактированию (если это необходимо).

Для автоматизированного проектирования технологических процессов изготовления электронной аппаратуры нашли применение ряд других пакетов прикладных программ. Например, TechAC, TECHCARD, Pro/ENGINEER и др.

## 8 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ САПР

*Информационное обеспечение САПР* - совокупность сведений, необходимых для выполнения автоматизированного проектирования в заданной форме.

*Основная функция ИО* - обеспечение создания, поддержки и организации доступа к данным.

### 8.1 Модели данных

1. *иерархическая,*
2. *сетевая,*
3. *реляционная.*

#### *Иерархическая модель данных*

Граф со следующим условиям:

- граф должен начинаться с одной вершины (корня);
- каждая вершина может иметь одну или несколько атрибутов (описывает конкретное свойство объекта);
- каждая вершина может быть связана с одним или несколькими порожденными узлами.

#### *Сетевая модель данных*

Отношение между данными представляются в виде сетей ( сетевого графа). В отличие от иерархической модели каждая запись (совокупность атрибутов) может находиться на любом уровне и вход в нее может быть реализован из любого узла.

#### *Реляционная модель данных*

Представляет собой набор двухмерных таблиц и связей между ними (в настоящее время наиболее часто встречается)

### **Типы таблиц и ключей в реляционных базах данных**

**Базовая таблица** – это таблица, которая включает один или несколько столбцов свойств объекта и которая содержит **первичный ключ**.

**Первичный ключ** состоит из набора значений атрибутов и однозначно определяет каждую строку или запись базовой таблицы. Ключ может быть простым и составным. Составной ключ включает в себя несколько атрибутов.

**Внешний ключ** – это столбец таблицы, значения которого соответствуют значениям первичного ключа, другой базовой таблицы.

**Промежуточная таблица** – это таблица, которая не объединяет свойств объекта или не содержит поле первичного ключа, служит для установки связей с другими таблицами.

**Условия**, накладываемые на реляционную модель:

1. Не должно быть одинаковых первичных ключей.
2. Все записи (строки) должны иметь одинаковую структуру в таблице.
3. Имена полей должны быть различимы, т.е. отличаться друг от друга.
4. Должна соблюдаться ссылочная целостность данных (т.е. каждому внешнему ключу должна соответствовать одна запись в базовой таблице).
5. Отношения в качестве компонентов не должны иметь другие отношения.

## 8.2 Нормализация данных

**Нормализацией** называется формальная процедура, в ходе которой атрибуты данных группируются в таблице, а таблицы группируются в базе данных.

**Задачи нормализации:**

- исключение повторений данных в таблицах и таблиц;
- создание структуры таблицы, которая предусматривает возможность дальнейшего изменения содержимого этой таблицы.

Нормализация выполняется по этапам (по нормальным формам).

### Первая нормальная форма

Для нее требуется, чтобы таблица была двумерной и не содержала повторяющихся групп данных.

Таблица 8.1 не приведена к первой нормальной форме, т.к. содержит повторяющиеся группы данных. Для приведения к первой нормальной форме, повторяющиеся группы данных нужно извлечь в новые таблицы 8.2 и 8.3.

Таблица 8.1

Судно	Название	№ маршрута	Отправление	Порт	Пункт	Прибытие	Отплытие	Пункт
137	Россия	2	18:10	Одесса	Севастополь	23:00	23:30	Керчь

Таблица 8.2 – Маршруты

Судно	Название	№ маршрута	Отправление	Порт
137	Россия	2	18:10	Одесса

Таблица 8.3 – Пункты

Пункт	Прибытие	Отплытие
Севастополь	23:00	23:30

Например, в Access атрибуты (столбцы) называются полями, а строчки называются записями.

## Связи в реляционной модели

Информация представляется в виде таблиц и связей между таблицами.

**Тип 1. Связь или отношение один к одному**, в таком отношении одной строке первой таблицы соответствует одна строка второй таблицы. Обычно такая связь используется для

- разбиения больших таблиц и
- предоставления различных прав доступа.

**Тип 2. Связь или отношение один ко многим**, одной строке первой таблицы соответствует несколько строк второй. Связь устанавливается с помощью первичного ключа первой таблицы, которая является базовой и внешнего ключа второй таблицы. *Это наиболее распространенный тип связи.*

**Тип 3. Связь или отношение многие к одному**. Этот тип противоположен типу 2.

**Тип 4. Связь или отношение многие ко многим**. Такое отношение нельзя представить в виде отношения между двумя записями двух таблиц. Это отношение задается с помощью промежуточной таблицы, связанной отношениями (многие к одному) с двумя базовыми таблицами. С помощью этой таблицы базовые таблицы будут связаны отношением многие ко многим.

## Вторая нормальная форма

Требуется, чтобы данные во всех неключевых полях полностью зависели от значений первичного ключа и каждого поля первичного ключа, если ключ является составным. Для приведения ко второй нормальной форме производится дальнейшее разбиение.

## Третья нормальная форма

Требуется, чтобы все неключевые поля таблицы зависели от первичного ключа, но не зависели друг от друга. Они предварительно должны быть приведены ко второй нормальной форме.

Обычно достаточно третьей нормальной формы для разработки эффективной структуры базы данных, хотя существуют еще четвертая и пятая нормальные формы.

## 8.3 Использование индексации таблиц

Индексы используются в СУБД для привязки полей таблиц к их физическому положению на диске.

**Основное назначение индекса** - ускорение доступа при выполнении запросов к таблице. Запрос позволяет извлекать или изменять информацию из таблиц или в таблице в соответствии с введенными критериями.

### **Общие рекомендации по использованию индексов.**

1. Индексируйте поля, которые часто используются в запросах на выборку данных.
2. Не индексируйте поля в таблицах, которые часто используются для до-

бавления и обновления записей.

**Целостность базы данных** складывается из двух элементов:

- целостность данных;
- ссылочная целостность.

Для обеспечения **целостности данных** требуется, чтобы значения первичного ключа таблицы были уникальны.

Для обеспечения **ссылочной целостности** требуется, чтобы каждому значению внешнего ключа соответствовало значение первичного ключа в базовой таблице.

Для обеспечения целостности данных в *Access* используются следующее:

1. **Тип данных** – счетчик, устанавливаемый для ключевого поля. Этот тип может иметь приращение на единицу для каждой следующей записи.
2. Индексирование поля первичного ключа со свойством **”да, совпадения не допускаются”**.

Поддержание **ссылочной целостности** должно исключать следующие транзакции (последовательность действий для изложения данных):

1. Добавление записи в таблицу со стороны «многие» в случае, если для данного внешнего ключа отсутствует запись со стороны «один».
2. Удаление таких же записей (нельзя удалять записи со стороны «один», если в связанной таблице со стороны «многие», есть соответствующие ей записи.).
3. Удаление записей в таблицах связанных отношением один к одному без соответствующих изменений в другой таблице.
4. Изменение значения поля внешнего ключа назначение таблицы.

## 8.4 Этапы разработки базы данных

1. Определить объекты (будущие таблицы).
2. Выявить связи между этими объектами.
3. На основе анализа свойств объекта установить перечень полей таблицы для соответствующего объекта.
4. Определить состав таблиц и связь между этими таблицами в базе данных.
5. Провести нормализацию таблиц.
6. Разработать структуру таблицы.

## 8.5 СУБД Access

**Назначение:** разработка базы данных и создание приложений базы данных с архитектурой «клиент-сервер».

### Режимы работы

1. **Режим запуска**, позволяет осуществить

- сжатие базы данных,
- преобразование и
- шифрование/дешифрование без открытия базы данных.

**2. Конструктор**, позволяет

- создавать и
- модифицировать структуру таблиц запросов, форм отчетов, макросов и модулей.

сов и модулей.

**3. Выполнение**, сама работа.

### Основные объекты Access

База данных *Access* содержит до **32756** следующих объектов:

1. **Таблица**. Одновременно может быть открыто до 1024 таблиц
2. **Запросы**. Запрос/Выборка может производиться одновременно не более чем из 16 таблиц, число полей в запросе не более 255.
3. **Формы**. Позволяет отображать и изменять данные в таблице или запросе в бале удобной для пользователя виде.
4. **Отчеты**. Используются для печати данных.
5. **Макросы**. Для автоматизации повторяющихся действий.
6. **Модули**. Содержат VBA код и предназначены для реакции на события в формах и отчетах, а также для реализации функций содержащихся в Access.

Отличие *Access* от большинства других СУБД состоит в том, что все хранится в одном файле.

### Таблицы

В режиме *конструктора* можно

- создавать и
- модифицировать структуру существующих таблиц.

Структура таблицы создается с помощью *свойств полей*, таких как

- **ключевое поле** предназначено для назначения первичного ключа таблицы и для выделения соответствующего поля;

- **имя поля** должно удовлетворять следующим требованиям: не более 64 символов включающих буквы, цифры, специальные символы, за исключением « . » « ! » « ' »;

- **тип данных** выбирается из раскрывающегося списка;

- **описание** является необязательным комментарием.

Остальные свойства полей приводятся на *вкладке общие*:

- **размер поля**;

- **формат поля** (задается формат вывода данных);

- **число десятичных знаков после запятой**;

- **маска ввода**

- **подпись** (псевдоним имени поля), если не писать, то в заголовке будет имя поля;

- **значение по умолчанию**;

- *условия назначения*;
- *обязательное поле*, указывает, требуется ли в данном поле обязательный ввод значений;
- *пустые строки*, определяет, допускается ли в данном поле пустая строка;
- *индексированное поле*;

### Типы данных

**МЕМО** – ограничение на длину текстового поля (255 символов). Используется для длинного текста, длина 65535 символов. Поле с типом данных **МЕМО не может быть**

- ключевым и
- индексироваться.

**Числовой** – для проведения вычислений, имеет *подтипы данных*:

- *Дата/Время*.
- *Денежный* – знак \$, RUB,DM.
- *Логический* – для обозначения факта.
- *Счетчик* – для соблюдения целостности данных.
- *Поле объектов OLE* – любой объект OLE.
- *Гиперссылка*.
- Мастер подстановок.

Установка связей между таблицами производится через схему данных (Сервис/Схема данных).

### Запросы

1. *Запрос на выборку*. Содержит данные из одной или нескольких таблиц, отображающиеся в связи с введенными критериями. Данные отображаются тоже в виде таблиц.

2. *Перекрестный запрос*. Отбирает данные из таблиц и представляет их в форме похожей на электронную таблицу.

3. *Запрос на изменение*. Позволяет объединить данные из разных таблиц.

4. *Запрос с параметрами*. Это запрос, свойства которого изменяются пользователем при каждом запуске.

### Формы

1. простая,
2. ленточная,
3. табличная.

Более подробно про СУБД Access будет рассмотрено в дисциплине «Автоматизированные системы обработки биомедицинской информации».

## РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

### Основная

1. Автоматизация проектирования радиоэлектронных средств: Учебное пособие для вузов / О.В. Алексеев, А.А. Головков, И.Ю. Пивоваров и др.: Под ред. О.В. Алексеева. - М.: Высш. шк., 2000. – 479 с.
2. Автоматизация машиностроения / Н.М. Капустин и др. – М.: Высш. шк., 2003. – 223 с.
3. САПР. Системы автоматизированного проектирования: Учеб. пособие для техн. вузов: В 9 кн. – Мн.: Выш. шк., 1989.
4. Ли К. Основы САПР (CFD/CAM/CAE). – СПб.: Питер Принт, 2004. – 560 с.
5. Разевиг В.Д. Проектирование печатных плат в P-CAD 2001. – М.: «СОЛОН-Р», 2001. – 557 с.
6. Федоренков А., Кимаев А. AutoCAD 2002: практический курс. – М.: «ДЕСС КОМ», 2002. – 576 с.
7. Система автоматизации технологического проектирования ТехноПро. Версия 5. Руководство пользователя. - М: АО «ТОП Системы», 2003. – 485 с.
8. Погорелов В. AutoCAD 2006: моделирование в пространстве для инженеров и дизайнеров. – СПб.: ВHV-СПб, 2006. – 368 с.
9. Автоматизация проектирования РЭС. Топологическое проектирование печатных плат: Учеб. пособие. – М.: Радио и связь, 2001. – 220 с.

### Дополнительная

10. Саврушев Э.Ц. P-CAD для Windows: система проектирования печатных плат: практическое пособие. – М.: Эком, 2002. – 320 с.
11. Тику Ш. AutoCAD 2004. – СПб.: Питер Принт, 2004. – 1040 с.
12. Ткачев Д. AutoCAD 2005. – СПб.: Питер Принт, 2005. – 462 с.
13. Мироненко И.Г. Автоматизированное проектирование узлов и узлов и блоков РЭС средствами современных САПР. - М.: Высш. шк., 2002. – 391 с.
14. Дендобренко Б.Н., Малика А.С. Автоматизация конструирования РЭА: Учебник для вузов. – М.: Высш. шк., 1980. – 384 с.
15. T-FLEX PARAMETRIC CAD. Двумерное проектирование и черчение. Трехмерное моделирование. Руководство пользователя. – М.: АО «ТОП Системы», 1999. – 245 с.
16. Techcard. Версия 3.5. Руководство пользователя / А.М. Куприянчик, И.М. Гинзбург, Ф.И. Печков и др. – Мн.: Репринт, 1999. – 183 с.
17. Аветисян Д.А. Автоматизация проектирования электрических систем. – М.: Высш. шк., 1998. – 254 с.
18. Кудрявцев Е.М. Mechanical Desktop Power Pack. Основы работы в системе. – М.: ДМК Пресс, 2001. – 544 с.
19. Pro/ENGINEER. Руководство по обучению основам конструирования. – USA: Parametric Technology Corporation. – 1996. - 243 с.
20. OrCad 10. Проектирование печатных плат / С.А. Кузнецова,

А.В. Нестеренко, А.О. Афанасьев; Под ред. А.О. Афанасьева. – М.: Горячая линия-Телеком, 2005. – 454 с.

21. Степанов Н.В., Голованов А.А. Практический курс пользователя Pro/Engineer 2001 / Под общ. ред. Д.Г. Красовского. – М.: КомпьютерПресс, 2001. – 271 с.

22. Разработка и оформление конструкторской документации РЭА: Справ. пособие / Э.Т. Романычева и др. – М.: Радио и связь, 1989. – 448 с.

### **Учебно-методические пособия**

23. Табаровец В.В., Стацук И.П. Методическое пособие к лабораторным работам по дисциплине «Системы автоматизированного проектирования». – Мн.: МРТИ, 1993. – 59 с.

24. Методическое пособие по разработке печатного монтажа / Ж.С. Воробьева, Н.С. Образцов, С.Н. Юрко и др. – Мн.: МРТИ, 1993. – 120 с.

25. Печатные платы в конструкциях РЭС: Учеб. пособие по курсу «Конструирование радиоэлектронных устройств» для студ. спец. Т 08.01 00 «Проектирование и производство РЭС» / Под ред. Ж.С. Воробьевой, Н.С. Образцова. – Мн.: БГУИР, 1999. – 87 с.

26. Станкевич А.В. Лабораторный практикум по дисциплине «Прикладные пакеты систем автоматизированного проектирования проблемно-ориентированных электронно-вычислительных средств». – Мн.: БГУИР, 2000. – 36 с.

27. Системы автоматизированного проектирования: Лаб. практикум для студ. спец. «Медицинская электроника», «Электронно-оптическое аппаратостроение» дневной формы обуч. В 3 ч. Ч. 1: Проектирование печатных плат в PCAD 2001 / В.М. Бондарик, А.М. Криштапович. – Мн.: БГУИР, 2004. – 63 с.

28. Системы автоматизированного проектирования: Лаб. практикум для студ. спец. «Медицинская электроника», «Электронно-оптическое аппаратостроение» дневной и заочной форм обуч.: В 3 ч. Ч. 2. Проектирование электронной аппаратуры в AutoCAD / В.М. Бондарик, С.В. Кракаевич, Н.Е. Казаринова, М.А. Беляцкий. – Мн.: БГУИР, 2005. – 53 с.

29. САПР P-CAD 2001: Лаб. практикум по курсу «Прикладные пакеты систем автоматизированного проектирования проблемно-ориентированных электронных вычислительных средств» для студ. спец. 40 02 02 «Электронные вычислительные средства» дневной формы обуч. / А.В. Станкевич, Д.С. Лихачёв. – Мн.: БГУИР, 2004. – 55 с.

30. Системы автоматизированного проектирования: Лаб. практикум для студ. спец. «Медицинская электроника», «Электронно-оптические системы и технологии» дневной и заочной форм обуч.: В 3 ч. Ч. 3. Параметрическое проектирование электронной аппаратуры в пакете T-FLEX CAD / В.М. Бондарик, С.В. Кракаевич, Д.В. Марковник. – Мн.: БГУИР, 2006. – 49 с.

31. Медицинская электроника. Дипломное проектирование / В.М. Бондарик, В.А. Бурский, А.Н. Осипов и др.; Под ред. А.П. Достанко. – Мн.: БГУИР, 2002. – 158 с.