

И. о. проректора по учебно-методической
работе В. В. Зубов

УТВЕРЖДАЮ



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ И ЗАДАНИЯ К КОНТРОЛЬНОЙ РАБОТЕ

Б1.В.09.01 ОСНОВЫ МЕХАТРОНИКИ И РОБОТОТЕХНИКИ

Специальность

21.05.04 Горное дело

Специализация

Мехатроника и робототехника промышленных комплексов

Одобрена на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой

(подпись)

Волков Е.Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического

(название факультета)

Председатель

(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

ОГЛАВЛЕНИЕ

Введение	4
Тема 1. Базовые определения мехатроники и робототехники	4
Тема 2. Общие тенденции развития мехатроники и робототехники	5
Тема 3. Технологическое обеспечение мехатронных и робототехнических систем.....	7
Тема 4. Современные требования к мехатронным и робототехническим модулям и системам.....	8
Тема 5. Концепция построения интеллектуальных мехатронных и робототехнических систем.....	9
Тема 6. Исполнительные модули мехатронных и робототехнических систем.....	11
Тема 7. Измерительно-информационные модули мехатронных и робототехнических систем.....	12
Тема 8. Модули управления мехатронными и робототехническими системами.....	13
Тема 9. Технологические мехатронные системы.....	14
Тема 10. Роботы и робототехнические системы.....	16
Тема 11. Транспортные мехатронные и робототехнические системы.....	18
Тема 12. Большие современные мехатронные системы различного назначения.....	20
Тема 13. Перспективные задачи и направления развития мехатроники и робототехники.....	21
Библиографический список.....	23

Введение

Предметом самостоятельной подготовки студентов в рамках учебного курса «Основы мехатроники и робототехники» являются мехатронные и робототехнические системы, их область применения и концепции их построения.

В основу курса положены современная теория и практика разработки мехатронных и робототехнических модулей и систем, включая собственные разработки автора.

Самостоятельная работа студентов включает в себя дополнительное самостоятельное изучение теоретического материала по темам рабочей программы с использованием рекомендуемой литературы и разъяснений к ней.

При работе с литературой студенты могут в индивидуальном темпе и стиле проводить собственный анализ изучаемого материала, систематизировать его, а благодаря зрительной памяти наиболее эффективно запомнить большее количество информации.

Самостоятельная работа студентов включает также подготовку к практическим занятиям (в соответствии с рабочей программой дисциплины) и подготовку к зачету и экзамену.

Проверка преподавателем самостоятельной работы осуществляется на консультациях, во время практических занятий, а также на зачете и экзамене.

Тема 1. Базовые определения мехатроники и робототехники

Изучение теоретического материала при помощи учебной литературы

В первую очередь необходимо познакомиться с различными интерпретациями понятий «мехатроника» и «робототехника», с соответствующими дополнительными комментариями к этим определениям.

Далее необходимо освоить базовые понятия мехатроники и робототехники: мехатронные модули движения, информационно-измерительные модули,

мехатронные модули систем управления, мехатронная машина и мехатронные процессы.

В заключение данной темы необходимо четко сформулировать основные отличия мехатронной машины от традиционной машины.

На самостоятельное освоение данной темы отводится 4 часа.

Подготовка к практическому занятию № 1

Практическое занятие включает знакомство с техническим оснащением кафедры «Мехатроника».

Контрольные вопросы по теме

1. Прокомментируйте основные понятия «мехатроника» и «робототехника». Укажите их общность и различие.
2. Сформулируйте цель, предмет и методы мехатроники и робототехники.
3. Сформулируйте определение термина «мехатронный модуль движения».
4. Сформулируйте определение термина «мехатронная машина».
5. Сформулируйте определение термина «информационно-измерительные модели».

Литература

[1] Лекция 2, с. 16–25.

[3] Глава 1, с. 16–22.

Тема 2. Общие тенденции развития мехатроники и робототехники

Изучение теоретического материала при помощи учебной литературы

Освоить понятие «синергетическое объединение» элементов мехатронных и робототехнических модулей и систем. Освоение понятий «интеграция», «интеллектуализация» и «миниатюризация», определяющих основные направления развития мехатронных и робототехнических систем.

От студентов требуется понимание пяти основных принципов организации интеллектуальных систем управления и основных направлений интеллектуализации мехатронных и робототехнических систем.

Студентам необходимо проследить основные этапы интеграции и миниатюризации мехатронных и робототехнических систем, а также иметь четкое представление о том, что все три направления развития мехатронных систем взаимосвязаны и влияют друг на друга.

На самостоятельное освоение данной темы отводится 4 часа.

Подготовка к практическому занятию № 2

Практическое занятие предполагает просмотр и обсуждение видеофильмов «Основы мехатроники», «Мобильные роботы».

Контрольные вопросы по теме:

1. Отличие мехатронных систем от традиционных механических и электромеханических управляемых систем.
2. Основные направления развития мехатронных систем.
3. Сформулируйте пять принципов организации интеллектуальных систем управления.
4. Требования, предъявляемые к функциональным характеристикам современных машин и комплексов.

5. В чем проявляется взаимосвязь трех основных направлений развития мехатроники и робототехники.
6. Историческая классификация мехатронных модулей по уровню синергетического объединения элементов модулей.

Литература

[1] Лекция 2, с. 25–34.

Тема 3. Технологическое обеспечение мехатронных и робототехнических систем

Изучение теоретического материала при помощи учебной литературы

Студентам необходимо освоить основные понятия, связанные со структурной и технологической пирамидами мехатроники: главные структурные части (механика, электроника, информатика) и базовые технологии мехатроники (гибридные технологии электромеханики, цифровые технологии управления движением, технологии автоматизированного проектирования).

При изучении базовых технологий мехатроники студентам необходимо понять концептуальную сущность указанных технологий и аппаратные средства для их реализации.

На самостоятельное освоение данной темы отводится 8 часов.

Подготовка к практическим занятиям № 3, № 4

Практические занятия предполагают применение конструкторов Lego и программных комплексов RoboLab для конструирования мобильных роботов.

Контрольные вопросы по теме

1. Назовите главные части мехатроники.

2. Назовите современные уровни развития базовых частей мехатронных систем.
3. Укажите три гибридные синергетические направления мехатронных систем.
4. Назовите главные базисные технологии мехатроники.
5. Охарактеризуйте современный уровень развития базисных технологий.
6. Назовите комбинированные технологии мехатроники.

Литература

[1] Лекция 3, с. 35–47.

[3] Глава 1, с. 70–86.

Тема 4. Современные требования к мехатронным и робототехническим модулям и системам

Изучение теоретического материала при помощи учебной литературы

Студенты должны понимать и формулировать стратегические, тактические и прикладные уровни требований к мехатронным и робототехническим модулям и системам. Проследить взаимосвязь и взаимообусловленность этих уровней требований.

Студенты должны изучить примеры систем, отвечающих прикладным требованиям, предъявляемым к мехатронным и робототехническим модулям и системам.

Студенты должны попытаться сформулировать дополнительные прикладные требования к мехатронным и робототехническим системам.

На самостоятельное освоение данной темы отводится 10 часов.

Подготовка к практическим занятиям № 5, № 6, № 7

Практические занятия предполагают изучение различных систем, отвечающих прикладным требованиям, предъявляемым к мехатронным и робототехническим модулям и системам.

Контрольные вопросы по теме

1. Сформулируйте стратегические требования к мехатронным и робототехническим системам.
2. Сформулируйте тактические требования к мехатронным и робототехническим системам.
3. Сформулируйте прикладные требования к функциональным и структурно-конструктивным показателям мехатронных и робототехнических систем.
4. Приведите примеры систем, отвечающих прикладным требованиям, предъявляемым к мехатронным и робототехническим системам.

Литература

[1] Лекция 4, с. 55–91.

Тема 5. Концепция построения интеллектуальных мехатронных и робототехнических систем

1. Изучение теоретического материала при помощи учебной литературы.

Студенты должны освоить определение интеллектуальных мехатронных и робототехнических систем, базу построения интеллектуальных систем управления (ситуационное управление и информационные технологии обработки знаний, обобщенную структуру интеллектуальных систем управления).

Кроме того, студенты должны изучить основные принципы проектирования мехатронных систем, познакомиться с общим алгоритмом проектирования

мехатронных и робототехнических систем (два этапа функционально-структурного подхода к проектированию мехатронных систем).

Студенты должны изучить основы построения экспертных систем. Познакомиться с основами автоматизированного проектирования мехатронных систем.

На самостоятельное освоение данной темы отводится 10 часов.

Подготовка к практическим занятиям № 8, № 9

На практических занятиях изучаются функциональные и структурные схемы интеллектуальных мехатронных и робототехнических модулей и систем, а также принципы построения и области применения экспертных систем.

Контрольные вопросы по теме

1. Сформулируйте признаки интеллектуальных систем управления, интеллектуальных мехатронных модулей и систем.
2. Какие современные информационные технологии используются в интеллектуальных системах управления?
3. Опишите основные блоки интеллектуальных систем управления.
4. Сформулируйте две основные идеи, на которых базируются интеллектуальные системы управления.
5. Сформулируйте понятие «совмещенное (параллельное) проектирование».
6. Объясните суть и значение для мехатроники синергетической интеграции элементов, входящих в состав системы.
7. Объясните суть модульного принципа проектирования мехатронных систем.
8. Чем объясняется широкое использование в мехатронных системах принципа перераспределения функциональной нагрузки от аппаратных модулей к информационным (компьютерным) модулям?

9. Опишите общий алгоритм проектирования мехатронных и робототехнических модулей.
10. Сформулируйте определение экспертной системы.
11. Укажите базовые функции экспертных систем.

Литература

[1] Лекция 3, с. 48–54; лекция 4, с. 94–100.

Тема 6. Исполнительные модули мехатронных и робототехнических систем

Изучение теоретического материала при помощи учебной литературы

Студенты должны изучить модули движения (мотор-редукторы, мотор-колеса, мотор-шпиндели, пьезоэлектрические модули, бионические модули, искусственную мышцу), мехатронные модули движения (безредукторный поворотный стол, автономный ортопедический аппарат), интеллектуальные модули движения (модуль для вальцовочных соединений, транспортный мобильный робот). Особое внимание необходимо обратить на конструкцию и системы управления различных исполнительных модулей.

На самостоятельное освоение данной темы отводится 4 часа.

Подготовка к практическому занятию № 10

На практическом занятии изучаются конкретные примеры модулей движения.

Контрольные вопросы по теме

1. Сформулируйте определения «модуль движения», «мехатронный модуль движения», «интеллектуальный модуль движения» и различия между этими модулями.

2. Принцип действия пьезоэлектрических приводов и бионических модулей движения.
3. Основные элементы интеллектуальных мехатронных модулей движения.
4. Классификация движителей мобильных систем.
5. Примеры модулей движения.

Литература

[1] Глава 3, с. 352–392.

Тема 7. Измерительно-информационные модули мехатронных и робототехнических систем

Изучение теоретического материала при помощи учебной литературы

Студенты должны изучить структурную и функциональную схемы передачи и обработки информации в мехатронных и робототехнических системах: усиления, нормирования (компандирования); фильтрация и преобразование в цифровую форму (дискретизация и квантирование сигнала по времени и кодирование); устройство связи с объектом; интерфейсы.

Далее студенты должны изучить измерительно-информационные модули различного назначения: механолюминесцентные сенсорные устройства сосредоточенного, распределенного и встроенного типов; двухкоординатный датчик силы микроманипулятора; скоростная путеобследовательская станция. При этом в первую очередь необходимо обратить внимание на принцип действия и структурные схемы измерительно-информационных модулей.

На самостоятельное освоение данной темы отводится 4 часа.

Подготовка к практическому занятию № 11

На практическом занятии изучаются примеры измерительно-информационных модулей различного назначения.

Контрольные вопросы по теме

1. Основные элементы измерительно-информационных модулей.
2. Типовая структурная схема измерительно-информационных модулей.
3. Основные типовые операции и преобразования информационных сигналов в измерительно-информационных модулях.
4. Примеры измерительно-информационных модулей.

Литература

[4] Глава 3, с. 439–457.

Тема 8. Модули управления мехатронными и робототехническими системами

Изучение теоретического материала при помощи учебной литературы

Студенты должны изучить особенности постановки задач управления в мехатронике. Принцип построения модулей управления. Иерархия управления в мехатронных системах. Степень интеллектуализации систем управления. Принципы построения интеллектуальных систем. Модули многоуровневых систем управления на исполнительном, тактическом и стратегическом уровнях (на различных слоях интеллектуальности).

Необходимо проанализировать природу (источники) возникновения неопределенностей, связанных с формированием управляющих воздействий (предсказуемые и непредсказуемые неопределенности). Рассмотреть системы

управления I и II рода в соответствии с характером неопределенностей в системе управления.

Изучить примеры модулей систем управления исполнительного уровня (адаптивное управление, управление с эталонной моделью, нейросетевое управление, нечеткое управление), тактического уровня (система управления движением робота), стратегического уровня (управление движением человека).

На самостоятельное освоение данной темы отводится 4 часа.

Подготовка к практическому занятию № 12

На практическом занятии изучаются системы управления мобильным роботом.

Контрольные вопросы по теме

1. Уровни иерархии управления мехатронными системами.
2. Системы управления I и II рода.
3. Источники неопределенности в мехатронных системах.
4. Четыре слоя обработки неопределенной информации (слои интеллектуальности).
5. Определение адаптивной системы управления.
6. Определение нечеткой системы управления.

Литература

[4] Глава 3, с. 458–467; с. 495–519.

Тема 9. Технологические мехатронные системы

Изучение теоретического материала при помощи учебной литературы

Студенты должны познакомиться с основными методами создания автоматизированных технологических мехатронных систем: технологическое обес-

печение автоматизированных систем. Разработка функционально-структурных схем систем, построение автоматизированных систем управления.

Затем студенты знакомятся с примерами построения мехатронных систем различного технологического назначения: процесс обжига окисленных окатышей, система вспомогательного кровообращения.

Далее студенты знакомятся с основными аспектами создания управления для различных способов производства ответственных изделий из титановых и высокопрочных алюминиевых сплавов для нужд различных отраслей машиностроения. Технологическими машинами с параллельной кинематикой (гексаподы); их конструктивными особенностями, преимуществами систем управления, сферами применения в металлообрабатывающей промышленности. Мехатронным станочным оборудованием с ЧПУ: пятью вариантами архитектурного решения систем ЧПУ (CNC, PCNC-1, PCNC-2, PCNC-3, PCNC-4); особенностями архитектурных решений, аппаратного уровня и программного обеспечения систем ЧПУ, контроллерами управления движением с открытой архитектурой как программной, так и аппаратной частей контроллера.

На самостоятельное освоение данной темы отводится 8 часов.

Подготовка к практическому занятию № 13

Практическое занятие предполагает изучение мехатронных металлообрабатывающих комплексов.

Контрольные вопросы по теме

1. Какими причинами определяется необходимость перехода к интеллектуальным системам управления кузнечнопрессовыми комплексами?
2. Опишите принципы управления процессом изотермического пресования на горизонтальных гидропрессах.

3. Опишите принципы управления процессом изотермической штамповки на вертикальных гидропрессах.
4. Опишите принципы управления процессом гидрорастяжения кольцевых заготовок на гидропрессе.
5. Опишите конструктивные особенности машин с параллельной кинематикой (гексаподов).
6. Укажите основные преимущества гексаподов перед другими технологическими машинами.
7. Укажите основные тенденции построения интеллектуальных контроллеров управления движением технических систем.
8. Какие задачи решает система ЧПУ?
9. Какие имеются архитектурные решения систем ЧПУ?
10. Что такое «открытая архитектура» систем ЧПУ?

Литература

[2] Глава 4, с. 9–47.

Тема 10. Роботы и робототехнические системы

Изучение теоретического материала при помощи учебной литературы

Студенты изучают краткую историю развития робототехники. Классификацию роботов по назначению, конструктивным признакам, способу управления, быстродействию и точности движений. Интеллектуальные робототехнические системы для бытового применения, медицинского обслуживания, досуга и развлечений, военного назначения. Промышленные робототехнические системы и комплексы: сборочные, технологические, кузнечно-прессовые, литья под давлением. Экстремальную робототехнику в промышленности, космосе, подводных роботах. Биоробототехнику, микроробототехнику: микросистемные

технологии, микроэлектромеханические системы, микророботы, микротехнологические модули.

В результате изучения данной темы у студентов должно выработаться целостное представление о роботах и робототехнических системах самого различного применения, принципах их построения.

На самостоятельное освоение данной темы отводится 8 часов.

Подготовка к практическому занятию № 14

На практическом занятии изучается шестистепенный учебный робот «Роботенок».

Контрольные вопросы по теме

1. Опишите краткую историю робототехники.
2. Приведите примеры удачной роботизации различных сфер деятельности человека.
3. По каким признакам производится классификация робототехники?
4. В чем состоит особенность построения и применения промышленных и робототехнических систем?
5. Рассмотрите перспективы специальной робототехники: космической, биологической, медицинской, экстремальной.
6. Назовите основные этапы развития микроробототехники.
7. Каковы перспективы развития микросистем?
8. Раскройте основные способы построения стационарных микросистем.
9. Какие типы приводов используют в мобильных микророботах?
10. Укажите основные области применения интеллектуальных роботов.

Литература

[2] Глава 4, с. 48–79.

Тема 11. Транспортные мехатронные и робототехнические системы

Изучение теоретического материала при помощи учебной литературы

Студенты изучают транспортные и робототехнические системы различного назначения. Железнодорожный транспорт: многофункциональный комплекс технических средств контроля состояния подвижного состава, инженерно-техническое оснащение высокоскоростного подвижного состава (вагоны, локомотивы, ходовые части подвижного состава, тормозное оборудование, устройства наклона кузова вагона). Инерционный накопитель энергии для тяговой цепи. Автомобильный транспорт: системы активной безопасности. Воздушный и водный транспорт: система автоматического управления самолетом (автопилот), дирижабли, автономные подводные аппараты. Нетрадиционные транспортные системы: транспорт с магнитным подвешиванием (магнитопланы), новая железнодорожная транспортная система (система автономных экипажей), трубопроводный транспорт.

На самостоятельное освоение данной темы отводится 8 часов.

Подготовка к практическим занятиям № 15, № 16

На практическом занятии изучается инженерно-техническое оснащение высокоскоростного железнодорожного транспорта и нетрадиционных транспортных систем.

Контрольные вопросы по теме

1. Укажите основные сферы применения мехатронных систем на железнодорожном транспорте.
2. Оцените уровень интеллектуальности железнодорожных систем (включая скоростной и высокоскоростной транспорт).

3. Укажите основные направления применения мехатронных систем в области локомотиво- и вагоностроения (включая рельсовые автобусы).
4. Оцените эффективность и перспективы применения накопителей энергии на железнодорожном транспорте.
5. Основные направления применения мехатронных систем на автомобильном транспорте.
6. Дайте общую характеристику систем активной безопасности движения автомобилей.
7. Какие существуют виды нетрадиционного транспорта?
8. Проанализируйте перспективы применения новой транспортной системы, предложенной в университете Падеборна (Германия).
9. Оцените перспективы высокоскоростного транспорта на магнитной подвеске.
10. Объясните принцип работы автопилота.
11. Каковы перспективы применения нейроконтроллеров в авиационном транспорте? Какими обстоятельствами диктуется необходимость их применения?
12. Для решения каких задач целесообразно применять дирижабли?
13. Сформулируйте тенденции развития дирижаблестроения.
14. В чем специфика управления дирижаблями?
15. Какие задачи призваны решать автономные подводные аппараты?
16. В чем специфика задач управления автономными подводными аппаратами?

Литература

[2] Глава 4, с. 80–164.

Тема 12. Большие современные мехатронные системы различного назначения

Изучение теоретического материала при помощи учебной литературы

Вначале студенты знакомятся с особенностями управления большими современными мехатронными системами. Далее студенты изучают примеры реализации управления большими современными мехатронными системами: коллективное управление группой роботов, дистанционное управление мобильными мехатронными и робототехническими системами посредством сети Интернет, макромоделью активного управления магистральными системами железнодорожного транспорта, интеллектуальной электроэнергетикой.

На самостоятельное освоение данной темы отводится 10 часов.

Подготовка к практическим занятиям № 17

На практическом занятии изучается система активного управления движением автомобильного и железнодорожного транспорта.

Контрольные вопросы по теме

1. Опишите общую структурную схему мобильных робототехнических систем.
2. Опишите обобщенную структуру управления робототехническими системами.
3. В чем состоит основная идея метода локального слежения за движущимися объектами?
4. Опишите структурную схему и основные элементы следящей системы технического зрения (ССТЗ).
5. Опишите обобщенную схему системы дистанционного управления мобильным роботом по сети Интернет.

6. Какие основные идеи закладываются в систему активного управления транспортной сетью (АУТС).
7. Объясните схему работы АУТС.
8. Основные элементы Европейской системы управления движением железнодорожного транспорта (ETCS).
9. Объясните основные аспекты построения и практического применения интеллектуальных сетей энергоснабжения.
10. Укажите ключевые различия между существующими в настоящее время и проектируемыми интеллектуальными сетями энергоснабжения.

Литература

[1] Лекция 4, с. 57–70; с. 100–132.

Тема 13. Перспективные задачи и направления развития мехатроники и робототехники

Изучение теоретического материала при помощи учебной литературы

Студенты должны сформулировать перспективные направления развития мехатронных и робототехнических систем: интеллектуализация систем управления технологическими комплексами различного назначения (металлообработка, транспорт, электроэнергетика), развитие микроэлектромеханических систем (микроробототехника, биоробототехника), создание самодостаточной системы интеллектуальных многофункциональных мехатронных машин для реконфигурируемого производства (построенного по принципу клеточного строения живых организмов из многофункциональных ячеек).

На самостоятельное освоение данной темы отводится 8 часов.

Подготовка к практическому занятию № 18

На практическом занятии рассматриваются перспективы развития микро-робототехники.

Контрольные вопросы по теме

1. В каком направлении должна продвигаться интеллектуализация управления технологических металлообрабатывающих комплексов?
2. За счет каких технических средств может осуществляться интеллектуализация робототехнических систем?
3. Какие виды транспортных систем нуждаются в модернизации?
4. Каковы перспективы развития микросистем?
5. Опишите новые служебные и функциональные задачи мехатроники и робототехники. Приведите примеры задач.
6. Дайте описание новых кинематических структур и конструктивных компоновок многокоординатных машин.
7. Поясните перспективы развития интеллектуальных мехатронных и робототехнических систем.

Литература

[1] Лекция 1, с. 9–15; с. 25–34.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Готлиб Б. М., Вакалюк А. А. Введение в специальность «Мехатроника и робототехника»: курс лекций. – Екатеринбург: Изд-во УрГУПС, 2012. – 134 с.
2. Готлиб Б. М. Введение в мехатронику: учеб. пособие: в 2 т. Т.2. Проектирование и применение мехатронных модулей и систем. – Екатеринбург: Изд-во УрГУПС, 2008. – 302 с.
3. Подураев Ю. В. Мехатроника: основы, методы, применение: учеб. пособие. – М.: Машиностроение, 2006. – 256 с.
4. Готлиб Б. М. Введение в мехатронику: учеб. пособие: в 2 т. Т.1. Концептуальные основы мехатроники. – Екатеринбург: УрГУПС, 2008. – 521 с.

Министерство науки и высшего образования РФ
ФГБОУ ВО
«Уральский государственный горный университет»

В. М. Таугер

**ДЕТАЛИ
МЕХАТРОННЫХ МОДУЛЕЙ**

Учебное пособие

Екатеринбург
2022

УДК 681.323-181.4(075)

Таугер В. М.

Детали мехатронных модулей: учебное пособие. – Екатеринбург: УГГУ, 2022. – 141 с.: ил.

Изложены методы расчета деталей преобразователей движения, необходимые для конструирования мехатронных модулей; рассмотрены конструктивные отличия основных деталей преобразователей движения мехатронных модулей от деталей передач общего назначения.

Учебное пособие предназначено для студентов направления «Мехатроника и робототехника промышленных производств».

Рецензент:

Рекомендовано редакционно-издательским советом университета.

ОГЛАВЛЕНИЕ

Введение.....	6
1. Соединения.....	7
1.1. Основные понятия.....	7
1.2. Шпоночные соединения.....	7
1.3. Зубчатые (шлицевые) соединения.....	10
1.4. Резьбовые соединения.....	13
1.4.1. Классификация резьбы.....	13
1.4.2. Геометрические параметры треугольной резьбы.....	13
1.4.3. Предотвращение саморазвинчивания в резьбовых соединениях.....	15
1.4.4. Расчет резьбовых соединений на прочность.....	15
1.4.5. Материалы и допускаемые напряжения.....	23
1.5. Заклепочные соединения.....	24
1.5.1. Разновидности заклепочных соединений.....	24
1.5.2. Расчет заклепочного соединения на прочность.....	26
1.6. Сварные соединения.....	28
1.6.1. Виды сварки.....	28
1.6.2. Соединения ручной электродуговой сваркой.....	30
1.6.3. Расчет сварных соединений на прочность.....	32
1.7. Соединения с натягом.....	36
Контрольные вопросы.....	36
2. Передачи.....	38
2.1. Общие сведения.....	38
2.2. Ременные передачи.....	40
2.3. Цилиндрические зубчатые передачи.....	43
2.3.1. Геометрические параметры цилиндрических зубчатых передач.....	43
2.3.2. Силы, действующие в зубчатом зацеплении.....	49
2.3.3. Виды разрушения зубьев.....	51
2.3.4. Материалы зубчатых передач.....	52

2.3.5. Методика расчета закрытой зубчатой передачи.....	53
2.4. Конические зубчатые передачи.....	73
2.4.1. Геометрические параметры конических зубчатых передач.....	73
2.4.2. Силы в конической зубчатой передаче.....	77
2.4.3. Методика расчета конической зубчатой передачи.....	77
2.5. Червячные передачи.....	85
2.5.1. Геометрические параметры червячных передач.....	85
2.5.2. Силы в червячной передаче.....	88
2.5.3. Материалы червячных передач.....	89
2.5.4. Методика расчета червячной передачи.....	90
2.6. Планетарные зубчатые передачи.....	96
2.6.1. Общие сведения о планетарных передачах.....	96
2.6.2. Передаточное число и условия существования планетарного механизма.....	99
2.6.3. Материалы планетарных передач.....	101
2.6.4. Методика расчета планетарной передачи.....	101
2.7. Волновые зубчатые передачи.....	105
2.7.1. Общие сведения.....	105
2.7.2. Передаточное число волновой передачи.....	107
2.7.3. Геометрические параметры волновой передачи.....	108
2.7.4. Материалы волновых передач.....	110
2.7.5. Методика расчета волновой передачи.....	110
Контрольные вопросы.....	119
3. Подшипники.....	121
3.1. Классификация подшипников по виду трения.....	121
3.2. Конструкции и классификация подшипников качения.....	122
3.3. Методика расчета подшипников качения.....	124
3.3.1. Исходные данные для расчета.....	124
3.3.2. Предварительный выбор подшипников.....	124
3.3.3. Проверочный расчет подшипников по динамической	

грузоподъемности.....	125
Контрольные вопросы.....	134
4. Валы и оси.....	136
4.1. Общие сведения.....	136
4.2. Методика расчета валов.....	136
4.2.1. Исходные данные.....	136
4.2.2. Проектировочный расчет.....	137
4.2.3. Разработка расчетной схемы.....	137
4.2.4. Построение эпюр изгибающих и крутящих моментов в сечениях вала.....	138
4.2.5. Проверка вала на усталостную прочность.....	138
Контрольные вопросы.....	139
Заключение.....	140
Библиографический список.....	141

ВВЕДЕНИЕ

Мехатронные модули представляют собой компактные устройства, всё более широко применяющиеся не только в различных отраслях промышленности, но и на транспорте, в медицине, в быту. Весьма перспективным представляется оснащение шахт, рудников и карьеров технологическими мехатронными модулями и системами, которые позволят не только повысить производительность предприятия, но и обеспечить безопасность труда и снизить вероятность возникновения аварийной ситуации.

Конструирование мехатронных модулей невозможно без знания методов расчета и конструирования их основных деталей и передач, называемых также преобразователями движения. В данном учебном пособии рассматриваются механические преобразователи движения, в качестве которых используют зубчатые (в том числе планетарные), червячные, волновые передачи.

В процессе разработки передачи конструктор должен учитывать ряд особых требований, предъявляемых к мехатронным модулям и решающим образом влияющих на выбор типа, расчет и конструкцию передачи.

Для надежного усвоения студентом материала пособия необходимы достаточно твердые и обширные знания дисциплин «Теоретическая механика» и «Сопротивление материалов».

1. СОЕДИНЕНИЯ

1.1. Основные понятия

Соединением называется неподвижное сопряжение деталей между собой.

По принципу *неразрушаемости при разборке* различают соединения разъемные и неразъемные. К *разъемным* относят соединения, допускающие разборку без разрушения или повреждения элементов, а к *неразъемным* – соединения, разборка которых невозможна без разрушения или повреждения элементов. В число элементов соединения включают как сами соединяемые детали, так и изделия или материалы, которыми данное соединение обеспечивается.

Из разъемных соединений наибольшее распространение получили следующие:

- шпоночные;
- зубчатые (шлицевые);
- резьбовые;
- соединения с некруглым валом.

Из неразъемных соединений наиболее часто используются:

- заклепочные;
- сварные;
- соединения с натягом (фрикционные);
- клеевые.

Соединения с некруглым валом и клеевые в настоящем учебном пособии рассматриваться не будут. Для их расчета следует обратиться к учебно-методической и технической литературе.

1.2. Шпоночные соединения

Шпоночные соединения образуются с помощью специальной крепежной детали, которая называется *шпонкой*. Шпонка располагается между соединяе-

мыми деталями и передает нагрузку (усилие или крутящий момент) с одной детали на другую.

Различают *напряженные* и *ненапряженные* шпоночные соединения. В первых напряжения в шпонке и соединяемых деталях возникают уже в процессе сборки, во-вторых же в отсутствие полезной нагрузки на соединение напряжения в шпонке и деталях практически равны нулю. В мехатронных модулях (ММ), как правило, применяются ненапряженные соединения.

Известны ненапряженные соединения круглой шпонкой, сегментной шпонкой и призматической шпонкой.

Круглая шпонка представляет собой цилиндрическую деталь (штифт), входящую по переходной посадке в отверстия соединяемых деталей. В большинстве случаев соединение служит для обеспечения точности сборки и не предназначено для передачи рабочей нагрузки.

Боковая поверхность *сегментной шпонки*, как показывает само название, представляет собой сегмент окружности. Соединение технологично в изготовлении и способно передавать небольшие нагрузки. Его расчет приведен в [1].

Наибольшие нагрузки могут быть реализованы в соединении *призматической шпонкой*. Используются шпонки со скругленными торцами, с одним скругленным торцом и с плоскими торцами (рис. 1.1, *а*). Размеры b , h и l оговорены ГОСТ 23360.

На валу 1 и во втулке 2 выполняются пазы, в которые при сборке устанавливается шпонка 3 (рис. 1.1, *б*), и крутящий момент T передается с вала к втулке ее боковыми гранями.

Боковые грани шпонки испытывают нормальные напряжения смятия, а в ее продольном сечении действуют касательные напряжения среза, но соотношение размеров стандартной низкой шпонки таково, что проверку на прочность по касательным напряжениям можно не выполнять.

Таким образом, расчет соединения на прочность производится по напряжению смятия, которое определяется по следующей формуле:

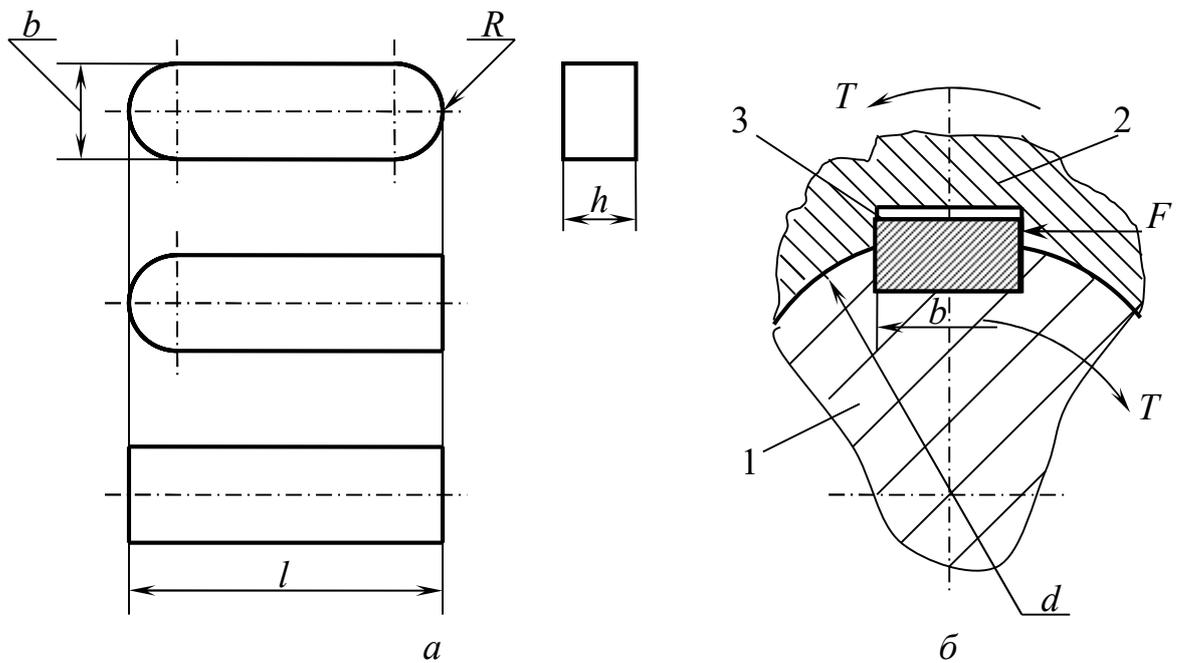


Рис. 1.1. Соединение призматической шпонкой

$$\sigma_{\text{см}} = \frac{F}{A} = \frac{4T}{dhl_p} \leq [\sigma_{\text{см}}], \quad (1.1)$$

где F – сила, действующая на боковую грань; A – рабочий участок боковой грани, по которому распределяется сила F ; l_p – рабочая длина шпонки; $[\sigma_{\text{см}}]$ – допустимое напряжение смятия.

Формула (1.1) получена с учетом следующих допущений:

- плечо силы F относительно центра сечения вала равно $d/2$;
- ширина участка боковой грани, на котором распределяется сила F , равна $h/2$.

Рабочая длина шпонки со скругленными торцами равна

$$l_p = l - b; \quad (1.2)$$

с одним скругленным торцом

$$l_p = l - \frac{b}{2}; \quad (1.3)$$

с плоскими торцами $l_p = l$.

Допускаемые напряжения $[\sigma_{см}] = 80...120$ МПа для соединений с переходными посадками втулки на вал, а для соединений с посадкой с натягом $[\sigma_{см}] = 110...200$ МПа. Меньшие напряжения принимают при чугунной втулке или при резких изменениях нагрузки, бóльшие – при стальной втулке и спокойной нагрузке.

1.3. Зубчатые (шлицевые) соединения

Соединения образуются сопряжением наружных зубьев на валу с внутренними зубьями в отверстии втулки.

По форме профиля зубьев различают соединения *прямобочные* (рис. 1.2, а), *эвольвентные* (рис. 1.2, б) и *треугольные* (рис. 1.2, в).

Соединения прямобочные выполняют с *центрированием* или по боковым сторонам зубьев, или по наружному диаметру, или по внутреннему диаметру вала. Центрирование по диаметрам обеспечивает более точную соосность соединяемых деталей, а центрирование по боковым сторонам – более точное распределение нагрузки между зубьями, т.е. передачу бóльших крутящих моментов. На рис. 1.2, а показано центрирование по наружному диаметру.

Эвольвентные соединения выполняют с центрированием или по боковым сторонам зубьев, или по наружному диаметру. На рис. 1.2, б показано центрирование по наружному диаметру. Эвольвентные шлицы можно получать на зуборезном оборудовании и достигать при этом высокой точности.

Размеры прямобочных соединений даны в ГОСТ 1139, эвольвентных – в ГОСТ 6033.

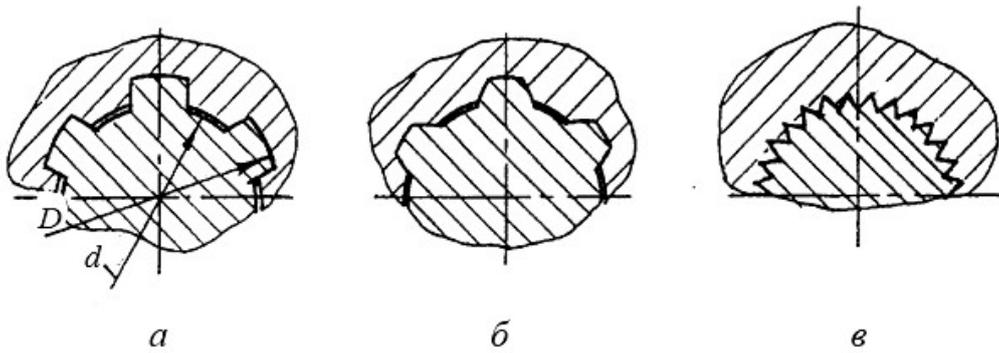


Рис. 1.2. Разновидности шлицевых соединений по форме профиля зуба

Треугольные шлицы не стандартизованы. Они применяются при тонкостенных втулках, а также для соединения пластмассовых деталей с металлическими валами.

С точки зрения применения в ММ шлицевые соединения обладают такими преимуществами перед шпоночными, как повышенные нагрузочная способность и точность центрирования деталей. К недостаткам можно отнести сложность изготовления.

Основными критериями работоспособности шлицевых соединений являются прочность на смятие и сопротивление коррозионно-механическому износу. Причина такого изнашивания в неподвижных соединениях заключается в микро-перемещениях сопряженных поверхностей при вращении вала.

При допущении равномерного распределения нагрузки между зубьями условие прочности по напряжениям смятия выглядит следующим образом:

$$\sigma_{\text{см}} = \frac{2T}{zhd_m l} \leq [\sigma_{\text{см}}], \quad (1.4)$$

где z – число зубьев; h – высота зуба; d_m – средний диаметр соединения; l – длина поверхности контакта.

Высота зуба и средний диаметр определяют по формулам

$$h = \frac{D-d}{2} - f'; \quad (1.5)$$

$$d_m = \frac{D+d}{2}, \quad (1.6)$$

где D , d – диаметры вершин и впадин зубьев соответственно; f' – расчетный зазор в соединении.

Допускаемое напряжение смятия определяют с учетом условий эксплуатации и твердости зубьев по табл. 1.1.

В случае постоянной нагрузки с числом циклов нагружения за полный срок службы порядка 10^8 условие удовлетворительного сопротивления соединения изнашиванию выражается неравенством

$$\sigma_{см} \leq [\sigma_{изн}]. \quad (1.7)$$

Таблица 1.1

Значения $[\sigma_{см}]$ шлицевых соединений

Условия эксплуатации	$[\sigma_{см}]$, МПа	
	Твердость зубьев <i>HB350</i>	Твердость зубьев <i>HRC40</i>
Тяжелые (с ударом)	26...38	30...52
Средние	45...75	75...105
Легкие	60...90	90...150

Условное допускаемое напряжение $[\sigma_{изн}]$ изменяется в широких пределах в зависимости от твердости поверхностей контакта и условий приложения нагрузки. Для улучшенных зубьев $[\sigma_{изн}] = 26...85$ МПа; для закаленных: до *HRC40* $[\sigma_{изн}] = 34...105$ МПа; до *HRC45* $[\sigma_{изн}] = 42...130$ МПа. При необходимости точного определения значений $[\sigma_{см}]$, $[\sigma_{изн}]$ следует обратиться к учебной и технической литературе.

1.4. Резьбовые соединения

1.4.1. Классификация резьбы

По форме основной поверхности различают цилиндрическую и коническую резьбу. Наиболее распространена цилиндрическая резьба. Коническую применяют для плотных соединений труб, масленок, пробок и т. п.

По форме профиля резьбового выступа различают треугольную, круглую, прямоугольную, трапецеидальную и упорную резьбу. Форма профиля тесно связана с назначением резьбы: для образования соединений используются треугольную и круглую (*крепежную*) резьбу, а в винтовых механизмах – прямоугольную, трапецеидальную и упорную (*ходовую*) резьбу. Такое распределение объясняется более высокой относительной прочностью крепежной резьбы и большими силами трения в соединении крепежной резьбой.

По направлению винтовой линии различают правую и левую резьбу. У правой резьбы винтовая линия идет слева направо и вверх, у левой – справа налево и вверх. Обычно применяют правую резьбу, левую – только в некоторых специальных случаях.

Различают *метрическую* и *дюймовую* треугольную резьбу. Геометрические параметры метрической резьбы выражены в миллиметрах, дюймовой – в долях дюйма.

Разновидность дюймовой резьбы – *трубная* резьба, резьбовые выступы и впадины которой скруглены. Соединения трубной резьбой имеют меньшие зазоры, чем соединения метрической резьбой, поэтому применяются в трубопроводной арматуре.

В ММ преимущественно применяется соединения треугольной резьбой, они и будут рассмотрены в дальнейшем.

1.4.2. Геометрические параметры треугольной резьбы

Основные геометрические параметры (рис. 1.3): α – угол профиля, для метрической резьбы $\alpha = 60^\circ$, для дюймовой резьбы $\alpha = 55^\circ$; d – наружный диаметр; d_1 – внутренний диаметр; d_2 – средний диаметр; p – шаг резьбы.

Шаг резьбы – расстояние между одноименными сторонами соседних профилей, измеренное в направлении оси резьбы. По величине шага различают резьбы с крупным шагом и с мелкими шагами. Крупный шаг для определенного d один, а мелких шагов несколько. С уменьшением шага соответственно уменьшаются размеры резьбового выступа и угол подъема витка (см. ниже).

По образующей воображаемого цилиндра, диаметр которого равен среднему диаметру резьбы, ширина резьбового выступа равна ширине впадины (и равна $b/2$).

Кроме того, выделяют такие параметры, как n – число заходов; $p_1 = np$ – ход резьбы, т.е. перемещение гайки по винту за один оборот; ψ – угол подъема витка.

Под углом ψ подразумевается угол подъема развертки винтовой линии по среднему диаметру:

$$\psi = \operatorname{arctg} \frac{p_1}{\pi d_2} = \operatorname{arctg} \frac{np}{\pi d_2}. \quad (1.8)$$

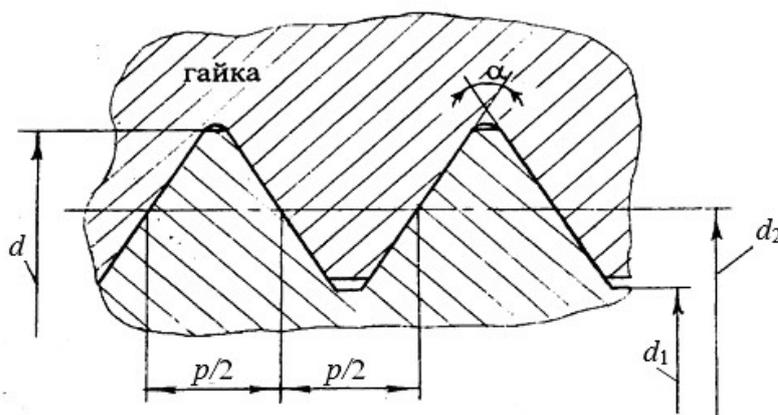


Рис. 1.3. Основные геометрические параметры треугольной резьбы

От величины ψ зависит, будет ли резьба *самотормозящейся*. Самоторможение – неперемное условие для крепежной резьбы, поскольку без его соблюдения соединение не в состоянии выдерживать осевую нагрузку. В однозаход-

ной треугольной резьбе $\psi = 2^\circ 30' \dots 3^\circ 30'$, что дает гарантированное самоторможение.

1.4.3. Предотвращение саморазвинчивания в резьбовых соединениях

Весьма часто резьбовые соединения эксплуатируются в условиях вибрации, переменных и ударных нагрузок. При этом обеспечения условия самоторможения недостаточно для предотвращения саморазвинчивания, т.к. вследствие переменного характера нагрузки силы трения в резьбе понижаются.

Существует много способов дополнительного *стопорения* резьбы. Способы первой группы направлены на повышение и стабилизацию сил трения в резьбе. Основные и наиболее часто применяемые из них – постановка *контргайки* и использование *пружинной шайбы*. Контргайка создает дополнительное натяжение, а, следовательно, и трение в резьбе, не зависящее от внешней нагрузки. Пружинная шайба представляет собой, по сути дела, виток пружины и поддерживает натяг и трение в резьбе на участке самоотвинчивания в один – два оборота гайки.

Способы второй группы сводятся к жесткому креплению элементов (гайки с болтом, гайки или винта с деталью). Одним из таких способов является применение в соединении специальной *корончатой гайки* со *шплинтом*. Корончатая гайка имеет кольцевой выступ с прорезями (коронку). Шплинтом называется деталь, изготовленная из проволоки полукруглого сечения. После навинчивания гайки на резьбовый стержень шплинт вставляется в прорезь коронки так, что проходит через коронку и резьбовый стержень насквозь (в стержне заранее сделано отверстие под шплинт). Затем концы шплинта отгибают, после чего шплинт надежно фиксирует гайку относительно резьбового стержня.

Указанными способами можно предотвратить саморазвинчивание в большинстве резьбовых соединений. В противном случае следует обратиться к литературе и подобрать приемлемый способ.

1.4.4. Расчет резьбовых соединений на прочность

Основные виды разрушения резьбовых соединений – срез витков и разрыв резьбового стержня. Касательные напряжения среза зависят, при равных диа-

метре и шаге резьбы, от количества витков резьбы, по которым распределяется нагрузка, т.е. от высоты гайки. Поэтому добиться равнопрочности резьбы и стержня винта можно подбором высоты гайки. Стандартная высота нормальной гайки $H \approx 0,8d$, и именно такая высота дает выполнение условия равнопрочности. Следовательно, при использовании в болтовом соединении гайки нормальной высоты исключается необходимость рассмотрения прочности витков, и расчет соединения сводится к *расчету стержня болта (винта)*.

Ниже рассмотрены распространенные в конструкциях ММ случаи нагружения резьбового стержня.

Случай 1. Стержень винта нагружен только внешней растягивающей силой F , затяжка соединения отсутствует.

Опасное сечение – по внутреннему диаметру резьбы. Условие прочности по напряжениям растяжения в стержне

$$\sigma = \frac{F}{\frac{\pi d_1^2}{4}} = \frac{4F}{\pi d_1^2} \leq [\sigma]. \quad (1.9)$$

Допускаемые напряжения $[\sigma]$ здесь и далее см. в табл. 1.2.

Случай 2. Внешняя нагрузка отсутствует, соединение затянуто.

Случай характерен для крепления ненагруженных герметичных крышек, люков и т.п.

Стержень болта растягивается осевой *силой затяжки* $F_{\text{зат}}$ и закручивается *моментом сил* T , необходимым для обеспечения затяжки. Величина $F_{\text{зат}}$ определяется из условия герметичности по рекомендациям, учитывающим опыт эксплуатации аналогичных соединений. Расчет стержня производится по *эквивалентному напряжению*, учитывающему наличие как нормальных напряжений растяжения, так и касательных напряжений кручения. Для стандартной метрической резьбы соотношение эквивалентного и нормального напряжений

выражается зависимостью $\sigma_3 \approx 1,3\sigma$, что позволяет рассчитывать стержень болта по формуле

$$\sigma_3 = \frac{1,3F_{\text{заг}}}{\frac{\pi d_1^2}{4}} = \frac{5,2F_{\text{заг}}}{\pi d_1^2} \leq [\sigma]. \quad (1.10)$$

Случай 3. Соединение нагружено силами, сдвигающими детали в стыке, болт поставлен в отверстия с зазором. Пример – соединение, показанное на рис. 1.4.

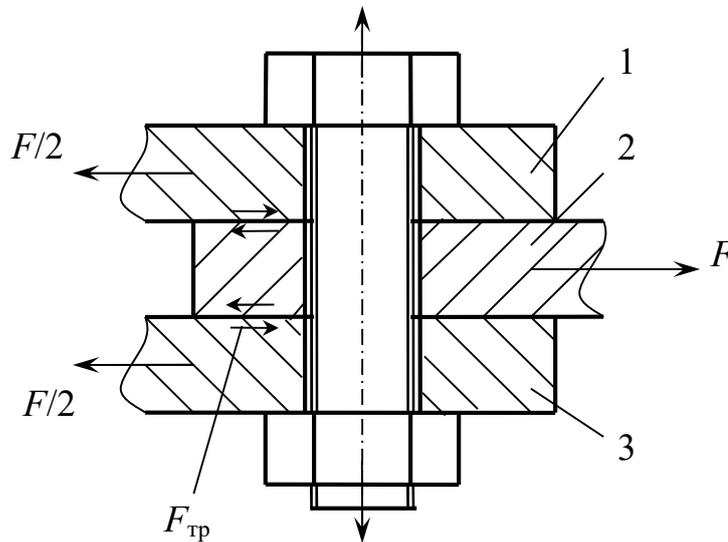


Рис. 1.4. Соединение, нагруженное поперечными силами (болт в отверстиях с зазором)

Для упрощения расчета принято следующее допущение: болт не касается своей боковой поверхностью стенок отверстий в деталях. Следовательно, сдвигу деталей препятствуют только силы трения в стыке. Условие отсутствия сдвига может быть получено из рассмотрения равновесия детали 2:

$$F = iF_{\text{тр}} = iF_{\text{заг}}f, \quad (1.11)$$

где i – число плоскостей стыка деталей; f – коэффициент трения скольжения в стыке (для стальных деталей $f = 0,15 \dots 0,2$).

Требованиям практики равенство (1.11) не удовлетворяет, т.к. малейшее увеличение силы F или уменьшение коэффициента трения (например, в результате попадания смазки в соединение) приведет к сдвигу деталей. Поэтому вместо (1.11) используется выражение

$$KF = iF_{\text{заг}} f, \quad (1.12)$$

откуда

$$F_{\text{заг}} = \frac{KF}{if}, \quad (1.13)$$

в (1.13) K – коэффициент запаса. При статической нагрузке $K = 1,3 \dots 1,5$, при переменной нагрузке $K = 1,8 \dots 2$.

После определения $F_{\text{заг}}$ прочность болта оценивают по (1.10).

Случай 4. Соединение нагружено силами, сдвигающими детали в стыке, болт поставлен без зазора. Такие соединения (см. рис. 1.5) образуются с помощью болтов по ГОСТ 7817, имеющих гладкую рабочую часть, диаметр которой d_3 больше диаметра резьбы d . Отверстия под болты обрабатывают разверткой, в результате посадки болтов в отверстиях получаются переходные или с натягом.

Сдвигающие силы вызывают в стержне болта напряжения среза и смятия. Резьба в соединении играет вспомогательную роль, фиксируя соединяемые детали одну относительно другой.

Напряжение среза в стержне болта в соединении по рис. 1.5

$$\tau = \frac{4F}{\pi d_3^2 i} \leq [\tau], \quad (1.14)$$

где i – число плоскостей среза (на рис. 1.5 $i = 2$).

Напряжение смятия для крайней детали

$$\sigma_{см1} = \frac{F}{2d_3\delta_1} \leq [\sigma_{см}]; \quad (1.15)$$

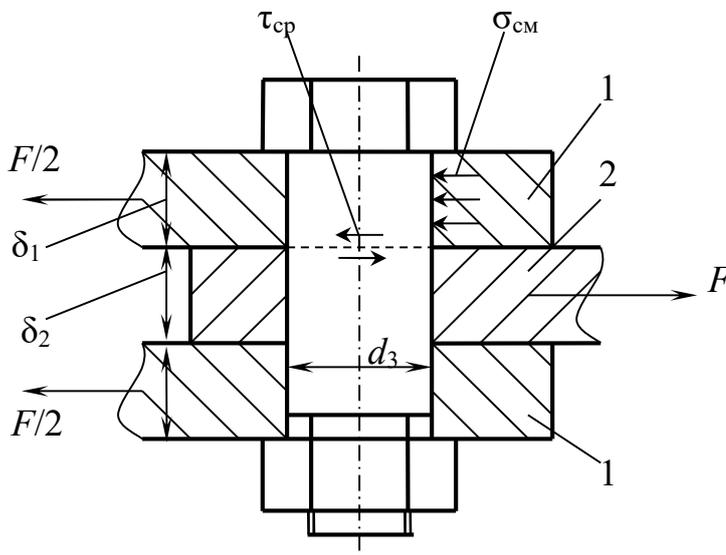


Рис. 1.5. Соединение, нагруженное поперечными силами (болт в отверстии без зазора)

для средней детали

$$\sigma_{см2} = \frac{F}{d_3\delta_2} \leq [\sigma_{см}]. \quad (1.16)$$

Формулы (1.15), (1.16) справедливы как для болта, так и для деталей. Из двух значений $\sigma_{см}$ расчет прочности выполняют по наибольшему, а допускаемое напряжение определяют по более слабому материалу болта или детали. Обычно диаметр болта находят из условия прочности на срез, а затем производят проверку по напряжениям смятия.

Случай 5. Болт затянут, внешняя нагрузка раскрывает стык деталей. В качестве примера могут быть рассмотрены болты крепления крышки резервуара к корпусу (рис. 1.6).

Внутри резервуара находится газ под избыточным давлением p . Очевидно, что затяжка болтов должна обеспечивать герметичность соединения, для чего болты предварительно (до того, как в резервуар подается газ) затягивают. При этом болты и стык деформируются: болты растягиваются, стык сжимается. Сжатие стыка происходит в основном за счет прокладки, если предусмотрена мягкая прокладка. Если же прокладка металлическая, то главную роль играет податливость фланцев крышки и корпуса. После того, как в резервуаре установилось давление p , приходящаяся на болт внешняя нагрузка становится равной

$$F = \frac{p\pi D^2}{4z}, \quad (1.17)$$

где z – число болтов.

Под действием внешней нагрузки дополнительно растягиваются. Но при этом крышка приподнимается болты, и сжатие стыка уменьшается на величину дополнительной деформации болтов. Таким образом, с одной стороны, имеет место приращение нагрузки на болт за счет силы давления газа на крышку, а с другой стороны, уменьшается нагрузка на болт со стороны стыка, возникшая в результате предварительной затяжки. В итоге суммарное увеличение нагрузки на болт оказывается значительно меньше, чем F по (1.17).

Расчетная суммарная нагрузка на болт

$$F_p = F_{\text{зат}} + \chi F, \quad (1.18)$$

где χ – коэффициент внешней нагрузки, обычно принимается равным 0,2...0,3.

Силу затяжки рекомендуется принимать

$$F_{\text{зат}} = k_{\text{зат}} F, \quad (1.19)$$

где $k_{\text{зат}}$ – коэффициент затяжки.

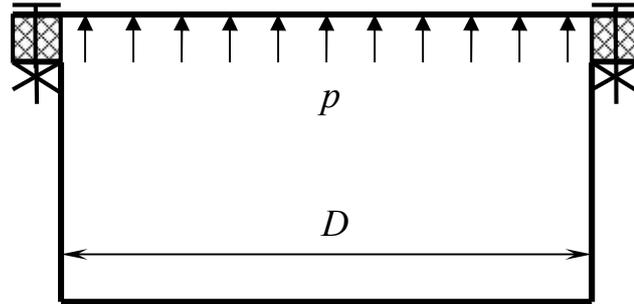


Рис. 1.6. Схема случая 5 резьбового соединения

По условию герметичности: при мягкой прокладке $k_{зат} = 1,3...2,5$; при металлической фасонной прокладке $k_{зат} = 2...3,5$; при металлической плоской прокладке $k_{зат} = 3...5$.

После того, как найдена F_p , проверяют болт на прочность по формуле

$$\sigma_3 = \frac{5,2F_p}{\pi d_1^2} \leq [\sigma]. \quad (1.20)$$

Случай 6. Эксцентрично нагруженный болт. Пример – нагружение болта с молотовидной головкой (рис. 1.7). Такие болты используют, когда невозможно расположить в отверстии обычный болт (отверстие слишком близко к стенке), а также в некоторых других случаях.

Затяжка соединения вызывает возникновение в стержне болта напряжений растяжения

$$\sigma_p = \frac{4F_{зат}}{\pi d_1^2} \quad (1.21)$$

и напряжений изгиба

$$\sigma_{\text{и}} = \frac{32F_{\text{зат}}e}{\pi d_1^3}, \quad (1.22)$$

где e – эксцентриситет силы затяжки.

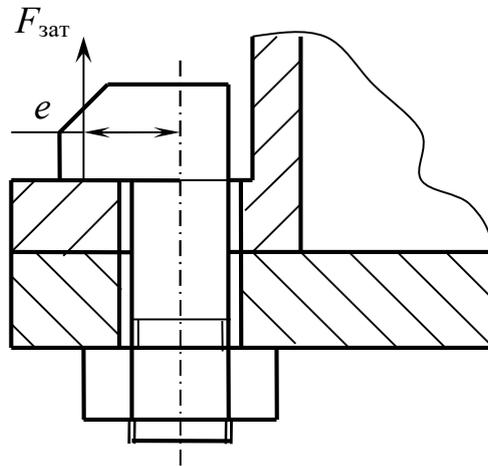


Рис. 1.7. Соединение болтом с молотовидной головкой

Для сопоставления величин составляющих напряжений предположим, что $e = d_1$. Тогда

$$\sigma_{\text{и}} = \frac{32F_{\text{зат}}}{\pi d_1^2}; \quad (1.23)$$

$$\frac{\sigma_{\text{и}}}{\sigma_{\text{р}}} = 8. \quad (1.24)$$

Из (1.24) видно, что из двух составляющих гораздо более опасно напряжение изгиба. Следовательно, эксцентричного нагружения болтов нужно всемерно избегать, а в тех случаях, когда использование эксцентрично нагруженного болта является технической необходимостью, обязательно учитывать его в расчетах.

Суммарное напряжение в стержне болта с учетом напряжения кручения

$$\sigma_p = \frac{5,2F_{\text{зат}}}{\pi d_1^2} + \frac{32F_{\text{зат}}e}{\pi d_1^3} \leq [\sigma]. \quad (1.25)$$

1.4.5. Материалы и допускаемые напряжения

Стандартные крепежные изделия изготавливают обычно из сталей марок Сталь 10...Сталь 35, так как эти дешевые стали позволяют выпускать большие количества изделий наиболее производительными методами. Стали с более высокими прочностными характеристиками применяют для изготовления высоконагруженных деталей в ответственных соединениях. В этих случаях может предусматриваться также термическая обработка.

Особое внимание следует уделять защите соединений от коррозии. С этой целью стандарты предусматривают больше десятка различных видов покрытий болтов, шпилек и гаек применительно к различным агрессивным средам – от цинкового с хромированием до серебряного. В тех случаях, когда вид среды неизвестен, можно рекомендовать достаточно простые и дешевые покрытия, такие, как кадмиевое с хромированием (группа 02).

В технически обоснованных случаях крепежные детали выполняют из цветных металлов и сплавов.

Допускаемые напряжения при расчете резьбовых соединений на прочность сведены в табл. 1.2.

Различают затяжку *контролируемую* и *неконтролируемую*. Контролируемая затяжка осуществляется с помощью специальных динамометрических ключей и ключей предельного момента, дающих возможность затянуть соединение заданной силой (и не большей). Существуют также и другие средства и методы контроля затяжки, к которым следует прибегать там, где это оговорено техническими требованиями. Судя по величинам запасов прочности (табл. 1.3), контролируемая затяжка позволяет существенно повысить надежность соединений.

Таблица 1.2

Допускаемые напряжения

Случай соединения	Номер формулы	Значение допускаемого напряжения
1	(1.9)	$[\sigma] = 0,6\sigma_T$
2,3,5	(1.10), (1.20)	$[\sigma] = \sigma_T/[s]$; [s] – по табл. 1.3 для неконтролируемой затяжки; $[s] = 1,5 \dots 2,5$ – для контролируемой затяжки
4	(1.14), (1.15), (1.16)	$[\tau] = 0,4\sigma_T$ – для статической нагрузки; $[\tau] = (0,2 \dots 0,3)\sigma_T$ – для переменной нагрузки; $[\sigma_{см}] = 0,8\sigma_T$ – сталь; $[\sigma_{см}] = (0,4 \dots 0,5)\sigma_T$ – чугун
6	(1.25)	$[\sigma] = 0,6\sigma_T$

Таблица 1.3

Запасы прочности при неконтролируемой затяжке

Материал болта	Запас прочности [s] для резьбы		
	M6...M16	M16...M30	M30...M60
Углеродистая сталь	5...4	4...2,5	2,5...1,5
Легированная сталь	6,5...5	5...3,3	3,3

1.5. Заклепочные соединения

1.5.1. Разновидности заклепочных соединений

Заклепочное соединение образуется расклепыванием стержня заклепки, вставленной в отверстия деталей (рис. 1.8). Обжимка 1 формирует замыкающую головку 2 заклепки 3, причем вследствие пластических деформаций стержень заклепки заполняет зазор в отверстиях. Поддержка 4 фиксирует закладную головку 5 заклепки.

Силы, вызванные упругими деформациями деталей и стержня заклепки, стягивают детали. Сдвигу деталей препятствует сопротивление стержня заклепки и частично силы трения между деталями.

Отверстия в деталях сверлят или продавливают. Сверление менее производительно, но придает соединению повышенную прочность. В ответственных соединениях предусматривается обязательное совместное сверление отверстий в деталях, что дополнительно повышает надежность соединения.

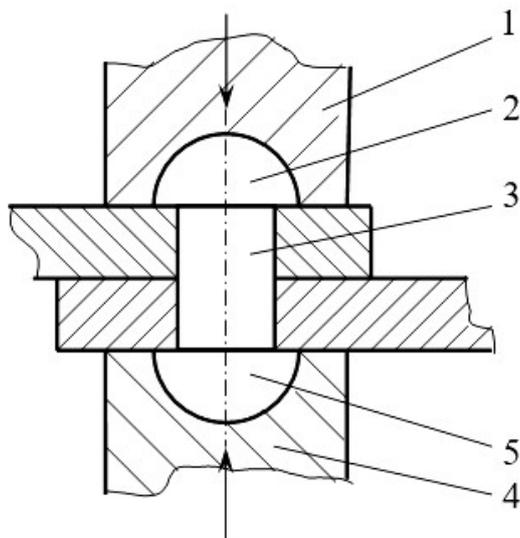


Рис. 1.8. Заклепочное соединение

Клепку можно производить вручную и машинным способом (пневматическими молотками, прессами и т. п.).

Стальные заклепки диаметром до 10 мм и заклепки из цветных металлов ставят без нагрева, поэтому процесс расклепывания называют холодной клепкой. Стальные заклепки большого диаметра ставят с нагревом. Нагрев повышает пластичность заклепки, облегчает расклепывание, улучшает заполнение отверстия и повышает натяг в соединении, связанный с температурными деформациями при остывании. В этом случае образование соединения называют горячей клепкой.

Применяются заклепки с полукруглой головкой (такая заклепка показана на рис. 1.8), с потайной и полупотайной головками. Кроме того, существуют различные типы специальных заклепок: пустотелые и полупустотелые, заклепки для односторонней клепки и т.д. Геометрическая форма и размеры заклепок нормальной точности оговариваются ГОСТ 10299, ГОСТ 10300, заклепок повышенной точности – ГОСТ 14787, ГОСТ 14798, ГОСТ 14801.

Листовые детали соединяются заклепочными швами. В зависимости от назначения различают швы *прочные*, *плотные* и *прочноплотные*. Прочные швы применяют в металлоконструкциях, плотные – в резервуарах для хранения

жидкостей и газов с невысоким давлением, прочноплотные – в резервуарах для хранения жидкостей и газов с высоким давлением. Плотность шва достигается постановкой заклепок с шагом, не большим некоторого строго определенного значения.

По конструктивному признаку различают швы *однорядные* и *многорядные*, соединения *внахлестку* и *встык*, *односрезные* и *многосрезные*. На рис. 1.9 приведен пример двухсрезного соединения *внахлестку*.

Применение заклепочного соединения целесообразно в тех случаях, когда материалы деталей плохо соединяются сваркой, а также в тех конструкциях, где важно растянуть во времени процесс разрушения.

К недостаткам соединения относятся трудоемкость выполнения длинных заклепочных швов, вредность работы клепальщика, существенное ослабление соединяемых деталей отверстиями под заклепки.

1.5.2. Расчет заклепочного соединения на прочность

На основные размеры заклепочных соединений выработаны нормы, по которым выбирают диаметры отверстия и заклепки, шаг шва и расстояние от шва до края деталей, а также толщину деталей. Расчет заклепки обычно носит проверочный характер.

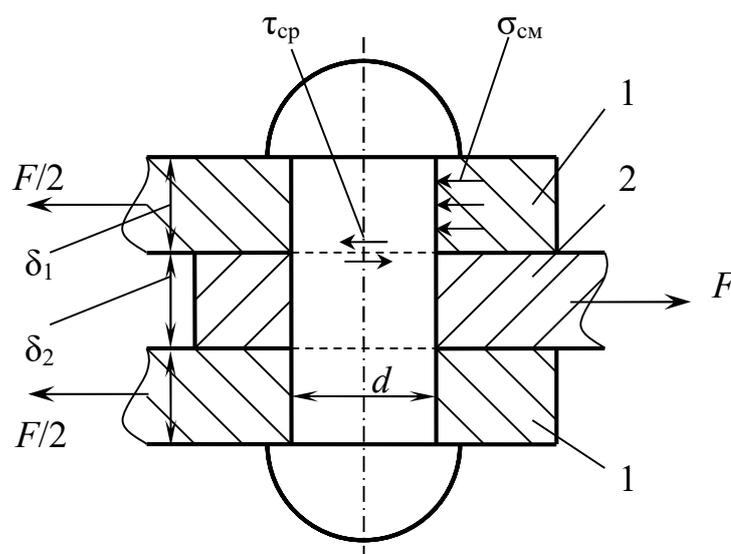


Рис. 1.9. Двухсрезное заклепочное соединение *внахлестку*

Рассмотрим соединение, нагруженное силами, сдвигающими детали в стыке (рис. 1.9). Его расчет аналогичен приведенному выше расчету резьбового соединения болтом, поставленным в отверстия без зазора, поэтому дополнительных пояснений не требует.

Условие прочности заклепки по напряжениям среза выражается формулой

$$\tau = \frac{4F}{\pi d^2 i} \leq [\tau_{\text{ср}}], \quad (1.26)$$

условия прочности по напряжениям смятия:

$$\sigma_{\text{см1}} = \frac{F}{2d \delta_1} \leq [\sigma_{\text{см}}], \quad (1.27)$$

$$\sigma_{\text{см2}} = \frac{F}{d \delta_2} \leq [\sigma_{\text{см}}]. \quad (1.28)$$

Допускаемые напряжения для заклепки из стали Ст3 принимают такими: $[\tau_{\text{ср}}] = 140$ (100) МПа; $[\sigma_{\text{см}}] = 320$ (280) МПа. Первые числа – для отверстий, полученных сверлением, значения в скобках – для отверстий, полученных продавливанием.

Материал заклепки должен отвечать следующим требованиям:

- обладать пластичностью;
- не принимать закалки при горячей клепке;
- не образовывать с материалом деталей гальванической пары.

1.6. Сварные соединения

1.6.1. Виды сварки

Из всего многообразия применяемых в настоящее время в производстве видов сварки при изготовлении ММ преимущественно используются *электродуговая* и *контактная*.

В электродуговой сварке образование соединения основано на заполнении стыка между деталями металлом *электрода*, расплавленного электрической дугой. Данный способ требует качественного прогрева кромок деталей и предотвращения доступа в зону сварки кислорода и азота воздуха. Последнее обеспечивается специальным покрытием электрода, которое, разлагаясь под действием температуры дуги, выделяет большое количество газа, инертного по отношению к металлу.

Различают *ручную* и *автоматическую* электродуговую сварку. Шов, выполненный сварочным автоматом, имеет более высокое качество, а следовательно, и большую статическую и усталостную прочность. К сожалению, выполнить соединение автоматически далеко не всегда возможно.

Контактная сварка является высокопроизводительным методом и применяется для соединения листовых деталей толщиной до 4 мм. Она основана на использовании повышенного электрического сопротивления зоны контакта деталей. Различают *точечную*, *шовную* и *стыковую* контактную сварку.

Сущность контактной сварки удобно пояснить на примере ее точечной разновидности (рис. 1.10).

Детали сжимаются электродами. Ток течет между электродами, при этом теплота в основном выделяется на поверхности контакта деталей; металл плавится, и образуется сварная точка.

Шовная сварка выполняется аналогично, но в качестве электродов применяют диски, которые перекатываются по деталям в направлении шва. Появляется возможность провести герметичный шов.

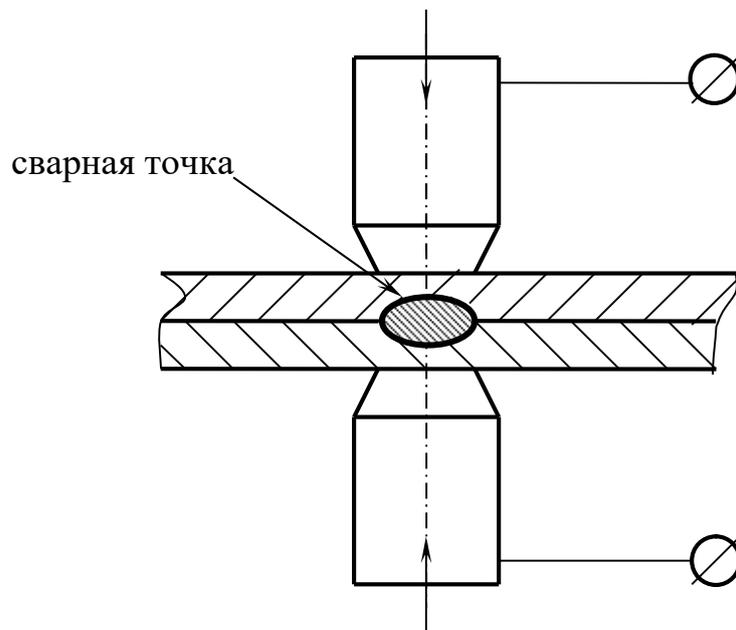


Рис. 1.10. Схема контактной точечной сварки

Стыковая контактная сварка применяется для соединения встык деталей типа стержней со сравнительно небольшой площадью поперечного сечения.

Сварные соединения обладают следующими достоинствами:

- высокая производительность и сравнительно невысокая трудоемкость сварки;
- простота обеспечения равнопрочности изделия, снижение его массы и стоимости.

Недостатки:

- необходимость правильного выбора материалов деталей;
- наличие в шве дефектов (неоднородностей, микротрещин и т.п.), и, как следствие, снижение прочности соединения.

Лучше всего свариваются детали из низкоуглеродистых сталей, например, из стали Ст 3. Стали углеродистые и легированные требуют для сварки применения специальных приемов: предварительного прогрева деталей, подачи инертного газа в зону сварки и т. д.

1.6.2. Соединения ручной электродуговой сваркой

Ручная электродуговая сварка представляет собой наиболее универсальный способ образования соединений, поэтому именно она и будет рассмотрена подробно.

Элементы сварных швов, получаемых ручной электродуговой сваркой, указаны в ГОСТ 5264. Стандарт устанавливает четыре типа соединений в зависимости от взаимного расположения соединяемых деталей: *стыковое, нахлесточное, тавровое и угловое*.

Стыковое соединение (рис. 1.11) простое и зачастую наиболее надежное. При толщине деталей $s \leq 6$ мм их можно соединять без *разделки кромок* (соединение С2). В случае $s > 6$ мм расплавленный металл электрода не может заполнить зазор между деталями, получается некачественный шов пониженной прочности. Поэтому при толщинах больших 6 мм применяют подварку с другой стороны, одностороннюю и двухстороннюю разделку кромок деталей (например, соединения С5 и С21).

Нахлесточное соединение (рис. 1.12) возникло как аналог заклепочного соединения внахлестку. Из всех сварных соединений оно наиболее простое, не требует подготовки кромок независимо от толщины деталей. Возможны одностороннее (Н1) и двухстороннее (Н2) нахлесточные соединения.

Тавровое соединение показано на рис. 1.13 и, подобно нахлесточному, может быть односторонним и двухсторонним. Кроме того, при больших s оно выполняется с разделкой кромок пристыковываемой детали.

Угловое соединение (рис. 1.14) часто применяется при изготовлении различного рода металлических емкостей – корпусов, коробов и т.п.

Различают два вида *швов*: *стыковой шов* – для образования стыковых соединений; *угловой шов* – для всех остальных соединений.

В обозначение типа *электрода* для ручной электродуговой сварки по ГОСТ 9467 входит буква «Э» и число, равное пределу прочности металла электрода, выраженному в кгс/мм², например, Э42, Э50А. Буква «А» в обозначении показывает, что химический состав электрода подвергается дополнитель-

ному контролю. Такие электроды применяются в ответственных соединениях для повышения надежности конструкции.

Стандарт устанавливает ряд диаметров электродов в миллиметрах: 2, 3, 4, 5, 6, 8, 10, 12... Для ручной сварки используют электроды небольших диаметров, как правило, до 6 – 8 мм, т.к. при этом достигается наивысшее качество шва в сочетании с невысокой трудоемкостью сварки.

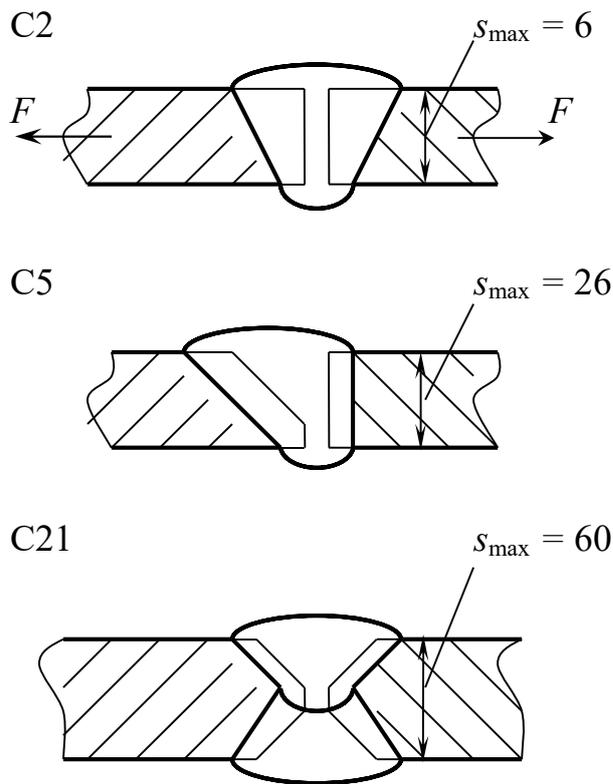


Рис. 1.11. Стыковое сварное соединение

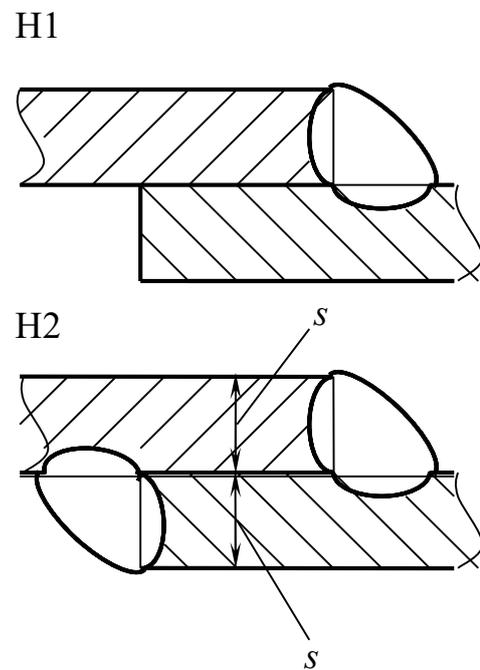


Рис. 1.12. Нахлесточное сварное соединение

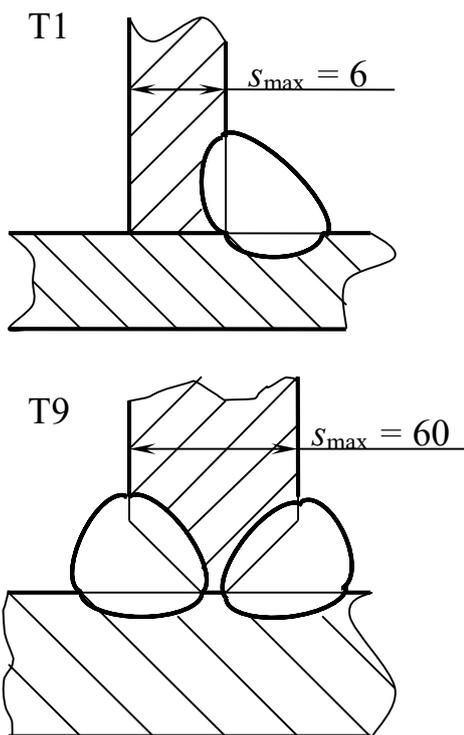


Рис. 1.13. Тавровое сварное соединение

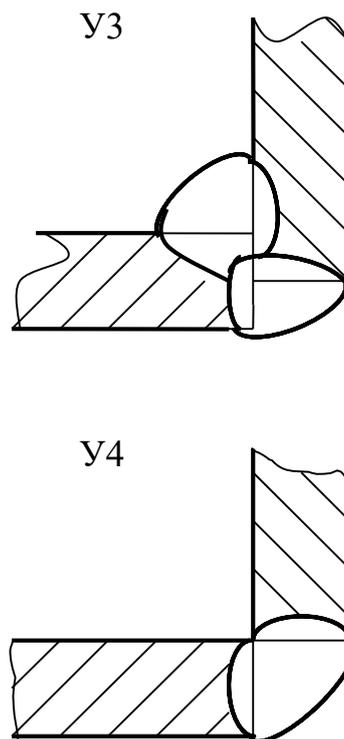


Рис. 1.14. Угловое сварное соединение

1.6.3. Расчет сварных соединений на прочность

Расчет стыкового шва производится следующим образом.

Напряжение в шве от растягивающей нагрузки (см. рис. 1.11) определяют по формуле

$$\sigma = \frac{F}{bs} \leq [\sigma'] = (0,9 \dots 1,0) [\sigma_p], \quad (1.29)$$

где b – длина шва; $[\sigma']$ – допускаемое напряжение для материала шва; $[\sigma_p]$ – допускаемое напряжение растяжения для материала деталей.

Предполагается, что стыковой шов практически равнопрочен с соединяемыми деталями. В (1.29) коэффициент 0,9 принимают при электродах Э42, Э50, а коэффициент 1,0 – при электродах Э42А, Э50А.

Напряженное состояние углового шва в нахлесточном и тавровом соединениях существенно отличается от напряженного состояния стыкового шва даже при простейшем нагружении растягивающими силами. В материале шва возникают как нормальные, так и касательные напряжения. Инженерный расчет производится упрощенно по касательным напряжениям. По форме швы разделяют на *нормальные* – в виде равнобедренного прямоугольного треугольника, *выпуклые* и *вогнутые*. Вогнутые швы лучше сопротивляются переменной нагрузке, но выполнение их связано с дополнительной механической обработкой, а следовательно, и с дополнительными затратами. В дальнейшем рассматриваются нормальные швы как самые распространенные в практике.

На рис. 1.15 показано нахлесточное соединение нормальным угловым швом с длиной L и катетом K . Разрушение такого шва происходит по биссектрисе AB прямого угла, что предсказано теорией и подтверждено практикой.

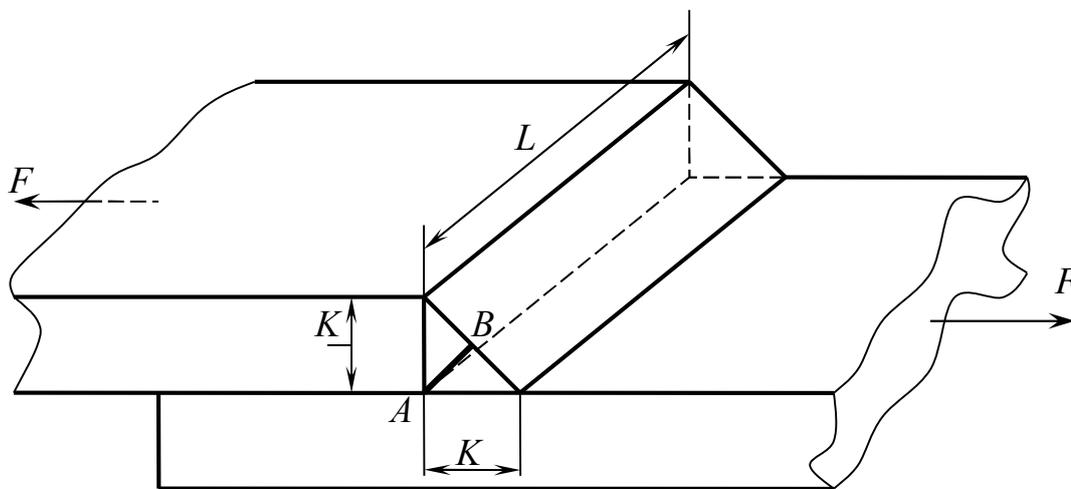


Рис. 1.15. Геометрия углового шва

Площадь опасного сечения шва

$$A_{o.c} = AB \cdot L \approx 0,7KL. \quad (1.30)$$

Условие прочности шва

$$\tau = \frac{F}{A_{o.c}} = \frac{F}{0,7KL} \leq [\tau']. \quad (1.31)$$

Допускаемое касательное напряжение для сварных швов, выполненных электродами Э42, Э50, принимают равным $[\tau'] = 0,6[\sigma_p]$, а для швов, выполненных электродами Э42А, Э50А – равным $[\tau'] = 0,65[\sigma_p]$.

По расположению относительно направления нагрузки различают швы *фланговые* (параллельные нагрузке), *лобовые* (перпендикулярные нагрузке) и *косые*. Напряжения в лобовом и фланговом швах различаются (при прочих равных условиях), но в инженерных расчетах касательные напряжения с достаточной степенью точности определяются по одним и тем же формулам.

На рис. 1.16 приведено соединение двумя фланговыми и одним лобовым швами. Для этого случая формула (1.31) принимает следующий вид:

$$\tau = \frac{F}{0,7K(2L_{\phi} + L_{л})} \leq [\tau'], \quad (1.32)$$

где L_{ϕ} , $L_{л}$ – длины флангового и лобового швов.

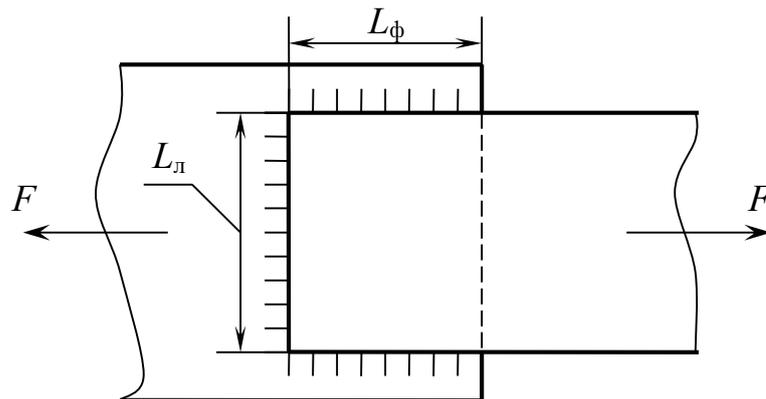


Рис. 1.16. Вариант нахлесточного соединения двумя фланговыми и одним лобовым швами

Соединения, показанные на рис. 1.17, нагруженные парой сил с моментом T , рассчитываются по следующим формулам:

- соединение на рис. 1,17, *a*

$$\tau = \frac{T}{0,7K L b} \leq [\tau']; \quad (1.33)$$

- на рис. 1.17, *б*

$$\tau = \frac{6T}{0,7K b^2} \leq [\tau']; \quad (1.34)$$

- на рис. 1.17, *в*

$$\tau = \frac{T}{0,7K L b + 0,7K \frac{b^2}{6}} \leq [\tau']. \quad (1.35)$$

В тех случаях, когда соединение находится одновременно под действием различных нагрузок (поперечная и продольная силы, крутящий момент и т. п.), расчет ведут, исходя из принципа независимости действия сил.

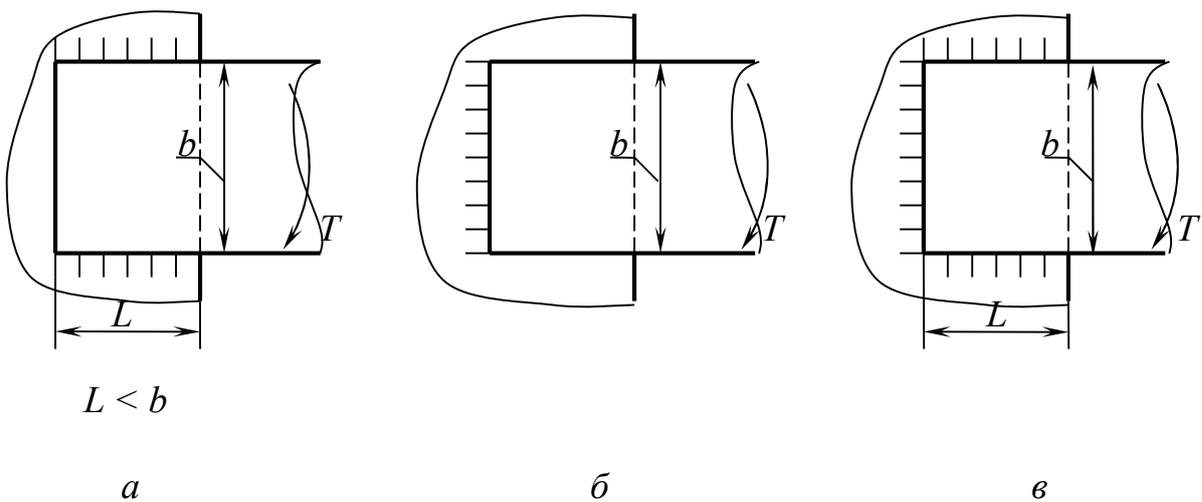


Рис. 1. 17. Соединения, нагруженные парой сил с моментом *T*

1.7. Соединения с натягом

Соединения типа «вал – ступица» с натягом (фрикционные) в ММ используются значительно чаще, чем в механизмах общего назначения, что объясняется их серьезными *преимуществами* по сравнению с прочими соединениями. Например, по сравнению со шпоночными соединения с натягом:

- имеют упрощенную технологию изготовления деталей;
- обеспечивают более точное базирование деталей в соединении;
- позволяют исключить элементы, снижающие прочность деталей, такие, как шпоночные пазы.

К *недостаткам* соединений с натягом относятся трудоемкость сборки и сложность контроля качества.

Основные способы сборки – сборка прессованием и сборка нагревом. При сборке нагревом нагревают ступицу (до температуры не выше 240 °С) или охлаждают вал (жидким воздухом до минус 190 °С, сухим льдом до минус 72 °С), что позволяет получать соединения, способные воспринимать бóльшие нагрузки, чем соединения прессованием.

Задача расчета соединения – правильный выбор посадки, натяг в которой обеспечит необходимое давление, а значит, и силы трения на поверхности контакта деталей. Методика расчета подробно рассмотрена в технической литературе.

Контрольные вопросы

1. Что называется соединением? Какие виды соединений используются в конструкциях ММ?
2. Как выполняется расчёт на прочность соединения призматической шпонкой?
3. Дайте сравнительную характеристику шпоночных и зубчатых соединений.
4. Укажите основные геометрические параметры метрической резьбы.

5. Опишите конструкцию болта в отверстие из-под развёртки.
7. Как выполняется расчёт на прочность эксцентрично нагруженного болта?
8. Как выполняется расчёт на прочность заклёпки?
9. Какие существуют стандартные разновидности соединений ручной электродуговой сваркой?
10. Укажите основные геометрические параметры углового шва.
11. Дайте сравнительную характеристику заклёпочных и сварных соединений.

2. ПЕРЕДАЧИ

2.1. Общие сведения

Механической передачей называется механизм, преобразующий параметры движения при его передаче от двигателя к исполнительным органам машины. Передача осуществляет согласование режима работы двигателя с режимом работы исполнительных органов. Применительно к ММ часто используется также термин *преобразователь движения*.

В ряде случаев передачи предназначены для изменения направления движения или для преобразования вращательного движения в поступательное или наоборот.

Часто в функцию передачи входит регулирование частоты вращения (скорости) исполнительного органа при постоянной скорости двигателя. Такая передача называется *вариатором*.

Основными параметрами движения являются *мощность* P_1 на входе и P_2 на выходе передачи и *частоты вращения* n_1 на входе и n_2 на выходе (либо *угловые скорости* ω_1 и ω_2 соответственно). Кроме того, различают производные характеристики:

- *коэффициент полезного действия* (КПД)

$$\eta = \frac{P_2}{P_1}, \quad (2.1)$$

- *передаточное отношение*, определяемое в направлении потока мощности,

$$i = \frac{n_1}{n_2} = \frac{\omega_1}{\omega_2}. \quad (2.2)$$

По величине i передачи делятся на понижающие при $i > 1$ и $n_1 > n_2$, или *редукторы*, и повышающие при $i < 1$ и $n_1 < n_2$, или *мультипликаторы*. В большинстве случаев частоту вращения требуется понизить, поэтому редукторы используются значительно чаще, чем мультипликаторы.

Кроме соотношений (2.1) и (2.2), в расчете передач часто используют следующие зависимости:

$$T = \frac{P}{\omega}, \quad (2.3)$$

$$\omega = \frac{\pi n}{30}; \quad (2.4)$$

$$T_2 = T_1 i \eta, \quad (2.5)$$

где T – крутящий момент на валу передачи; T_2, T_1 – крутящий момент на выходном и на входном валу соответственно.

Механические передачи делятся на *передачи трением* (фрикционные, ременные) и *передачи зацеплением* (зубчатые, червячные, цепные, винтовые). Передачи зацеплением по сравнению с передачами трением обладают повышенной нагрузочной способностью (или меньшими размерами при равной мощности), обеспечивают высокую точность и большую величину передаточного отношения, могут использоваться в широком диапазоне скоростей.

К недостаткам их можно отнести сложность изготовления, шум при высоких скоростях, неспособность компенсировать динамические нагрузки (жесткость).

Достоинствами передач зацеплением обуславливается их преимущественное использование в ММ. Поэтому далее о передачах трением дано только общее представление, а передачи зацеплением рассмотрены подробно.

2.2. Ременные передачи

Ременная передача (рис. 2.1) состоит из двух *шкивов* – ведущего 1 и ведомого 2, закрепленных на валах, и *ремня* 3, охватывающего шкивы.

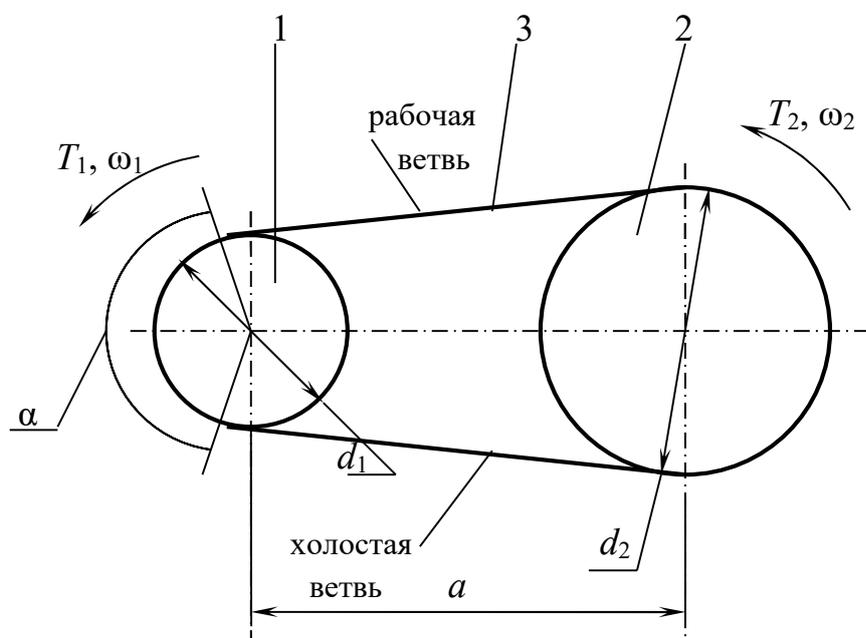


Рис. 2.1. Схема ременной передачи

Нагрузка передается силами трения, возникающими между ремнем и шкивами вследствие натяжения ремня. Передача также может включать в себя устройство для обеспечения требуемой силы натяжения ремня (*натяжное устройство*).

По форме поперечного сечения ремня различают передачи *плоскоременные* (рис. 2.2, а), *клиноременные* (рис. 2.2, б), *поликлиноременные* (рис. 2.2, в), *круглоременные*, а также передачи *пленочными ремнями*. Преимущественное распространение имеют передачи первыми двумя видами ремней.

Клиноременная передача по сравнению с плоскоременной имеет важные преимущества:

- бóльшие силы трения ремня по шкиву при равных силах натяжения, а следовательно, передача бóльших крутящих моментов и мощностей;

- передача осуществляется, как правило, несколькими клиновыми ремнями, в результате чего повышается ее надежность (выход из строя одного ремня еще не означает остановку механизма, а тем более, аварию).

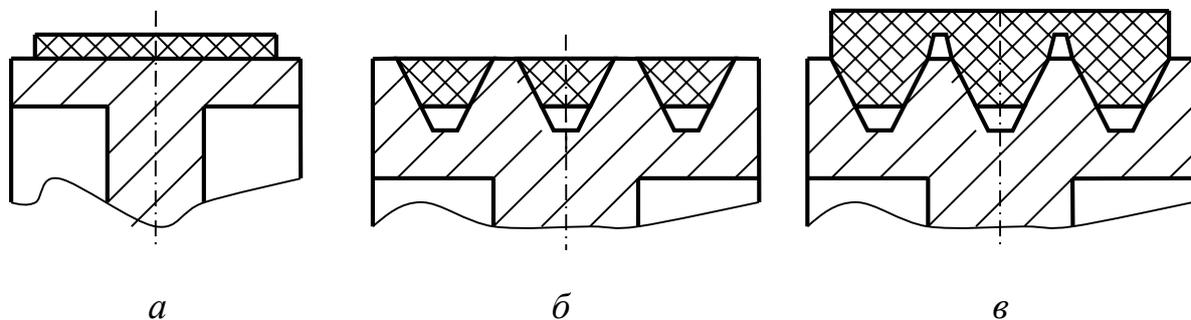


Рис. 2.2. Формы поперечного сечения ремней

К преимуществам плоскоремной следует отнести:

- возможность обеспечения значительных межосевых расстояний (размер a на рис. 2.1);
- возможность создания передач с непараллельными осями шкивов.

По способу натяжения ремней различают передачи с *натяжением при сборке*, с *периодическим подтягиванием* и с *автоматическим поддержанием натяжения*.

Способ натяжения ремня при сборке передачи самый простой: ремень с усилием надевают на шкивы и подтягивание его по мере износа и неупругой вытяжки не предусматривают. Нагрузочная способность такой передачи понижается, т.к. со временем натяжение ремня ослабевает.

Более совершенной в конструктивном отношении является передача, в которой возможно периодическое подтягивание ремня. Как правило, это достигается перемещением одного из шкивов, чаще – ведущего, с последующим закреплением на новом месте.

Передача с автоматическим поддержанием необходимого натяжения обычно содержит устройство в виде натяжного (плоскоремная передача) или

оттяжного (клиноременная передача) ролика, воздействующего на холостую ветвь ремня. Поджатие ролика к ремню обеспечивается пружиной или грузом.

Установка оттяжного ролика отрицательно сказывается на долговечности ремней, поэтому в клиноременных передачах чаще используют периодическое подтягивание.

К основным геометрическим параметрам ременной передачи относятся межосевое расстояние a , диаметры шкивов d_1 и d_2 , длина ремня L , угол обхвата ремнем малого шкива α .

Рекомендуемые величины a :

- для плоскоремennых передач

$$a \geq 2(d_1 + d_2); \quad (2.6)$$

- для клиноременных передач

$$0,55(d_1 + d_2) \leq a \leq 2(d_1 + d_2), \quad (2.7)$$

где h – высота сечения ремня.

Точное значение передаточного отношения ременной передачи

$$i = \frac{n_1}{n_2} = \frac{d_2}{d_1(1-\varepsilon)}, \quad (2.8)$$

где ε – коэффициент упругого скольжения ремня по шкиву.

При нормальной работе передачи $\varepsilon = 0,01 \dots 0,03$.

Соотношение крутящих моментов на шкивах

$$T_1 = \frac{T_2}{i\eta}, \quad (2.9)$$

где η – КПД передачи.

Плоскоременные передачи имеют $\eta \approx 0,97$, клиноременные – $\eta \approx 0,96$.

Методики расчета ременных передач изложены в учебной литературе.

2.3. Цилиндрические зубчатые передачи

2.3.1. Геометрические параметры цилиндрических зубчатых передач

Цилиндрические зубчатые передачи внешнего и внутреннего зацепления относятся к преобразователям вращательного движения с *параллельными осями колес*. Частным случаем такой передачи считают *реечное зацепление*, в котором одно из колес имеет бесконечно большой радиус, вследствие чего вырождается в прямолинейную *зубчатую рейку*.

По форме профиля зуба различают передачи *эвольвентные*, *циклоидные* и *Новикова*. В зацеплениях эвольвентном и циклоидном боковые стороны профиля зуба очерчены соответственно по эвольвенте и циклоиде. В зацеплении Новикова профиль зуба образован дугами окружностей.

Циклоидное зацепление применяется в кинематических передачах приборов. Наибольшее распространение получило эвольвентное зацепление: оно позволяет создавать достаточно прочные и малогабаритные преобразователи движения и обладает существенными технологическими преимуществами. Наиболее высокие прочностные характеристики имеет зацепление Новикова, однако оно значительно сложнее в изготовлении. Далее будут рассматриваться передачи с эвольвентными зубьями.

По расположению зубьев на колесах различают цилиндрические передачи *прямозубые* (зуб расположен по образующей цилиндра), *косозубые* (зуб расположен по винтовой линии) и *шевронные* (рис. 2.3).

Меньшее зубчатое колесо пары (рис. 2.4) называется *шестерней*, большее – *зубчатым колесом* (или просто *колесом*). Параметрам шестерни присваивается индекс 1, параметрам колеса – индекс 2.

Термины, определения и методы расчета геометрических параметров зубчатых передач стандартизованы.

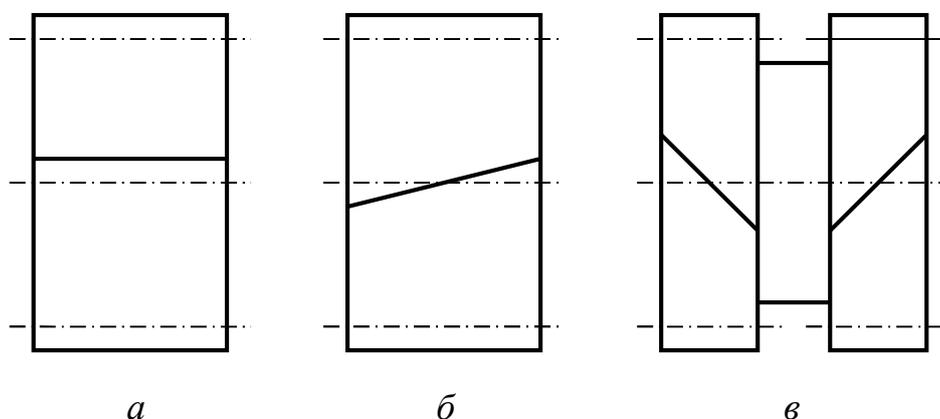


Рис. 2.3. Разновидности цилиндрических зубчатых колес по расположению зубьев:
a – прямозубое; *б* – косозубое; *в* – шевронное

Рассмотрим сначала прямозубую передачу, а затем – особенности геометрии косозубой передачи.

Числа зубьев – z_1 и z_2 .

Передаточное отношение от шестерни к колесу, называемое *передаточным числом*, равно

$$u = \frac{z_2}{z_1} . \quad (2.10)$$

Делительный окружной шаг зубьев p есть расстояние между сходственными точками двух соседних зубьев по дуге делительной окружности.

Делительной окружностью называется окружность, по которой производится деление заготовки на зубья. По дуге делительной окружности толщина зуба равна толщине впадины (и равна $\frac{p}{2}$).

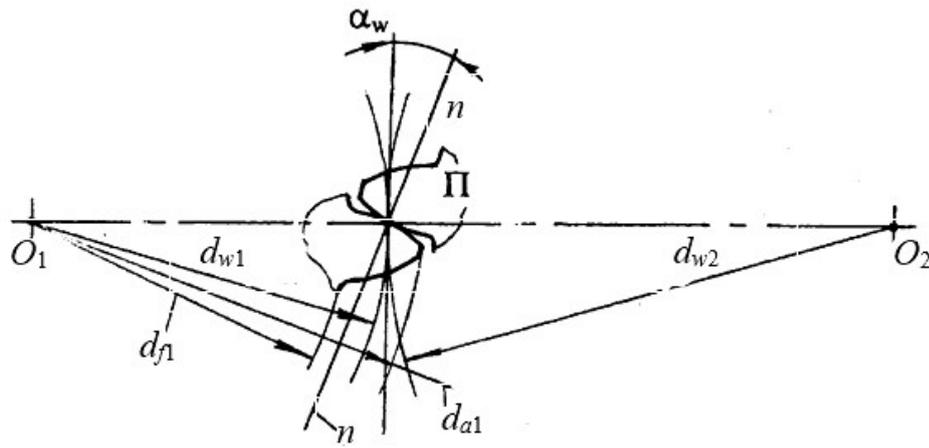


Рис. 2.4. Основные геометрические параметры
прямозубой передачи

Основной характеристикой размеров зубьев является *модуль*:

$$m = \frac{P}{\pi}. \quad (2.11)$$

Значения модулей указаны в ГОСТ 9563.

Высота зуба для шестерен и колес без смещения исходного контура (о смещении исходного контура см. ниже):

$$h = 2,25m. \quad (2.12)$$

В прямозубой передаче:

- диаметры делительных окружностей

$$\begin{aligned} d_1 &= mz_1; \\ d_2 &= mz_2; \end{aligned} \quad (2.13)$$

- диаметры окружностей вершин зубьев

$$\begin{aligned}d_{a1} &= m(z_1 + 2); \\d_{a2} &= m(z_2 + 2); \end{aligned} \quad (2.14)$$

- диаметры окружностей впадин

$$\begin{aligned}d_{f1} &= m(z_1 - 2,5); \\d_{f2} &= m(z_2 - 2,5); \end{aligned} \quad (2.15)$$

- межосевое расстояние

$$O_1O_2 = a_w = 0,5m(z_1 + z_2). \quad (2.16)$$

Кроме того, выделяют *начальные окружности*, по которым шестерня и колесо обкатываются в процессе вращения. Диаметры начальных окружностей равны

$$\begin{aligned}d_{w1} &= \frac{2a_w}{u + 1}; \\d_{w2} &= 2a - d_{w1}. \end{aligned} \quad (2.17)$$

Для передач без смещения $d_1 = d_{w1}$; $d_2 = d_{w2}$.

Точка касания начальных окружностей, обозначенная буквой П на рис. 2.4, называется *полюсом зацепления*.

Общая нормаль *n-n* к контактирующим поверхностям зубьев, проведенная через точку П, называется *линией зацепления*. При вращении шестерни и колеса точки контакта зубьев находятся на линии *n-n*. Угол между линией зацепления и перпендикуляром к *линии центров* O_1O_2 – *угол зацепления* α_w . Стандартная величина $\alpha_w = 20^\circ$.

На рис. 2.5 показано расположение двух соседних зубьев косозубого колеса. Сечения зубьев тремя плоскостями – нормальной *n-n*, торцовой *t-t* и осевой

$a-a$ – дают соответственно *нормальный модуль* m_n , *торцовый модуль* m_t и *осевой модуль* m_a . В нормальном сечении профиль косоугольного зуба совпадает с профилем прямого зуба, поэтому m_n должен быть стандартным. В торцовом и осевом сечениях модули зависят от угла наклона зуба β , стандарт на них не распространяется. В расчётах косозубых передач индекс n в обозначении нормального модуля по умолчанию не приводят.

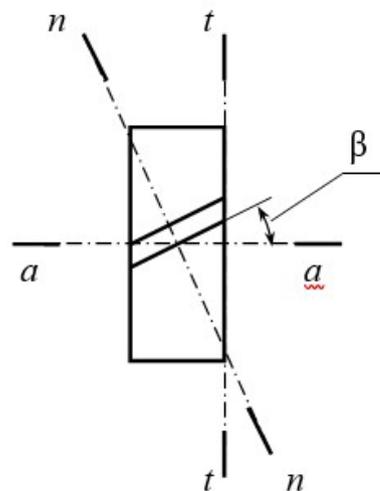


Рис. 2.5. Сечения зубьев косозубого колеса нормальной, торцовой и осевой плоскостями

Особое значение имеют геометрические параметры в торцовом сечении:

- модуль торцовый

$$m_t = \frac{m_n}{\cos \beta}; \quad (2.18)$$

- диаметр делительный

$$d = m_t z = \frac{m_n z}{\cos \beta}; \quad (2.19)$$

- шаг окружной

$$p_t = \frac{P_n}{\cos\beta}. \quad (2.20)$$

Геометрические параметры зубчатой передачи не исчерпываются указанными выше. Подробнее геометрия зубчатых передач изложена в учебной и технической литературе.

Рассмотрим, как число зубьев влияет на их форму.

Одним из наиболее технологичных и широко применяемых способов изготовления зубчатых колес является так называемый *способ обкатки*. Суть способа сводится к тому, что зубонарезающий инструмент в виде зубчатой рейки или шестерни вводится «в зацепление» с заготовкой, и перемещения инструмента и заготовки в процессе обработки подобны перемещениям пары деталей, находящихся в зацеплении.

При изготовлении обкаткой боковые стороны профиля зуба получаются эвольвентными. С увеличением числа зубьев колеса боковые стороны приближаются к прямолинейным, и в предельном случае, когда $z = \infty$ (зубчатая рейка), профиль приобретает форму равнобокой трапеции. Наоборот, с уменьшением числа зубьев толщина зуба у основания и вершины уменьшается, кривизна профиля увеличивается. Когда z становится меньше некоторого минимального значения z_{\min} , зубья инструмента, проворачиваясь во впадине заготовки, удаляют материал из ножки зуба. Это явление называется *подрезанием ножки*, оно существенно снижает прочность зуба. Для прямозубых передач считают $z_{\min} = 17$.

При необходимости выполнения $z < z_{\min}$, а также в некоторых других случаях применяют смещение нарезающего инструмента: инструмент отодвигают от положения, соответствующего нарезанию без смещения, на расстояние xm , где x – коэффициент смещения исходного контура. Смещение считают положительным, если оно направлено от центра, и отрицательным, если к центру

заготовки. Шестерни нарезают при положительном смещении, что позволяет существенно изменить форму зуба: он становится короче и толще, подрезание ножки устраняется.

Нарезание зубьев со смещением является определенным усложнением процесса изготовления, поэтому рекомендуется по возможности не назначать z меньше z_{\min} . Увеличения числа зубьев практически всегда можно достичь уменьшением модуля.

2.3.2. Силы, действующие в зубчатом зацеплении

На рис. 2.6 показано прямозубое зацепление.

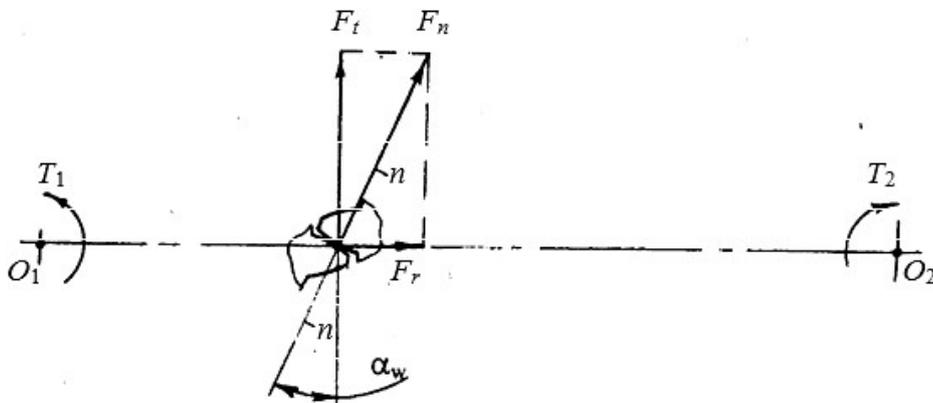


Рис. 2.6. Силы в прямозубом зацеплении

Силу нормального давления зуба шестерни на зуб колеса F_n можно разложить на две составляющих:

- окружную силу

$$F_t = \frac{2T_2}{d_2}; \quad (2.21)$$

- радиальную силу

$$F_r = F_t \operatorname{tg} \alpha_w. \quad (2.22)$$

Саму силу F_n вычисляют по выражению

$$F_n = \frac{F_t}{\cos \alpha_w}. \quad (2.23)$$

В косозубом зацеплении (рис. 2.7) нормальную силу раскладывают на три составляющих:

- окружную силу – см. формулу (2.21);
- радиальную силу

$$F_r = \frac{F_t \operatorname{tg} \alpha_w}{\cos \beta}; \quad (2.24)$$

- осевую силу

$$F_a = F_t \operatorname{tg} \beta. \quad (2.25)$$

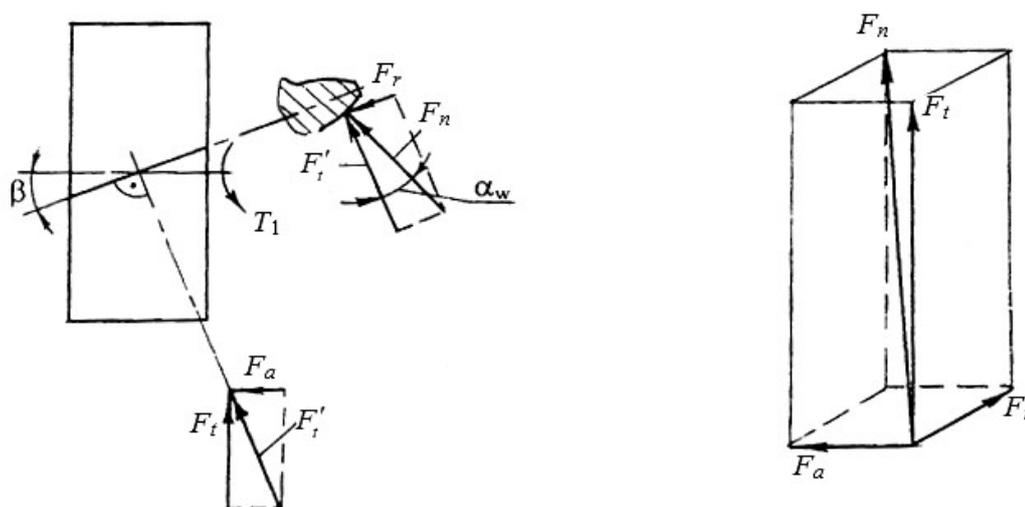


Рис. 2.7. Силы, действующие на зуб косозубого колеса

Нормальная сила является диагональю параллелепипеда с ребрами, равными F_t , F_r и F_a , и может быть найдена так:

$$F_n = \frac{F_t}{\cos \alpha_w \cos \beta}. \quad (2.26)$$

2.3.3. Виды разрушения зубьев

Основной вид разрушения поверхности зубьев при хорошей смазке передачи, надежно защищенной от попадания пыли и грязи, – *усталостное выкрашивание*. Зубья таких передач разделены тонким слоем масла, устраняющим металлический контакт. Износ зубьев мал. Передача работает длительное время до появления усталости в поверхностных слоях зубьев. На поверхности появляются небольшие углубления, которые растут и превращаются в раковины. Выкрашиванию способствует смазка, она запрессовывается зубьями в трещины и своим давлением отделяет частицы металла. Причина усталостного выкрашивания – контактные напряжения σ_H .

Основные меры предупреждения выкрашивания: расчет передачи на усталость по контактным напряжениям; применение материалов с повышенной твердостью поверхности; повышение точности изготовления и монтажа передач. Во многих случаях выкрашивания можно избежать, если предусмотреть приработку зубьев в процессе эксплуатации передачи (о приработке см. ниже).

Поломка зубьев связана с напряжениями изгиба σ_F . Различают *полный* (по всей длине) и *угловой* излом. Одна из причин полного излома – перегрузки ударного или статического характера. Другая причина – усталостная поломка от действия переменных напряжений. Причиной углового излома являются погрешности передачи, в результате которых нагрузка воспринимается не всей длиной зуба, а концентрируется на одном из его углов.

Поломку от перегрузок предупреждают защитой передачи посредством различных предохранительных устройств или учетом перегрузок при расчете; поломку от переменных напряжений предупреждают определением размеров из

расчета на усталость. К общим мерам относятся: увеличение модуля, положительное смещение при нарезании зубьев, термообработка, устранение концентраторов напряжений (рисок от обработки, раковин в отливках, микротрещин от термообработки). Углового излома можно избежать повышением точности сборки передачи (в частности, обеспечением параллельности осей колес) и применением зубьев со срезанными углами.

Открытые передачи, а также закрытые, но недостаточно защищенные от попадания абразивных частиц, выходят из строя в основном из-за *абразивного износа*. По мере износа зубьев увеличиваются зазоры в зацеплении, появляется шум, возрастают динамические нагрузки. Толщина зубьев уменьшается, и соответственно снижается их прочность.

Основные меры борьбы с износом: повышение твердости поверхности зубьев, защита от загрязнения, применение специальных смазочных материалов. Большое значение имеет своевременное диагностирование сверхнормативного износа и замена изношенных колес.

Кроме перечисленных видов разрушения зубьев, наблюдаются такие, как заедание, пластический сдвиг, отслаивание твердого поверхностного слоя. Однако при грамотном расчете, качественном изготовлении и правильной эксплуатации передачи вероятность этих разрушений значительно ниже.

2.3.4. Материалы зубчатых передач

В настоящее время установлено, что контактная прочность зубьев определяется в основном твердостью материала. Наибольшая твердость, а следовательно, наименьшие габариты и массу передачи можно получить при изготовлении колес из сталей, подвергнутых термической обработке.

В зависимости от твердости стальные зубчатые колеса разделяют на две основные группы: твердостью $HB < 350$ и твердостью $HB > 350$.

Твердость $HB < 350$ позволяет производить чистовое нарезание зубьев после термообработки, в результате чего можно получать высокую точность без дорогих отделочных операций (шлифовки, притирки и т.п.). Колеса этой группы хорошо прирабатываются и не подвержены хрупкому разрушению при динами-

ческих нагрузках. Под *приработкой* понимают износ поверхностей зубьев передачи, приводящий к более равномерному распределению нагрузки по длине зуба, а потому существенно повышающий надежность зубчатых колес. Приработку зубьев широко используют в условиях индивидуального и мелкосерийного производства в мало- и средненагруженных передачах. Обычно для лучшей приработки твердость шестерни назначают на 20...50HB больше, чем твердость колеса.

Твердость материалов второй группы ($HB > 350$) обычно выражают в единицах *HRC*. Соотношение единиц *HB* и *HRC* таково: $1HRC \approx 10HB$. Специальными видами термообработки могут быть получены твердости 50...60 *HRC*, причем допускаемые контактные напряжения возрастают примерно в два раза, а нагрузочная способность передачи – в четыре раза по сравнению с передачей, изготовленной из материалов первой группы. Очевидно, что применение высокотвердых материалов является большим резервом повышения нагрузочной способности зубчатых передач.

К недостаткам материалов этой группы следует отнести плохую приработываемость и, как следствие, необходимость повышенной точности изготовления деталей передач и их монтажа. Кроме того, некоторые виды термообработки (объемная закалка, цементация) сопровождаются значительным короблением зубьев. Для исправления формы зубьев требуются дополнительные операции.

Данные по механическим характеристикам некоторых наиболее широко используемых сталей приведены в табл. 2.1. Материалы группы $HB < 350$ представлены нормализованными и улучшенными сталями, а группы $HB > 350$ – закаленными объемной или поверхностной закалкой, а также азотированными.

В зависимости от способа получения заготовки различают литые, кованные, штампованные колеса и колеса из круглого проката.

2.3.5. Методика расчета закрытой зубчатой передачи

2.3.5.1. Общие положения

Настоящая методика основана на ГОСТ 21354 и ГОСТ 2185, предназначена для расчета на усталостную и статическую прочность эвольвентных цилин-

дрических зубчатых передач и может быть использована студентами при конструировании ММ в ходе курсового и дипломного проектирования.

Таблица 2.1

Механические характеристики сталей

Группа сталей	Марка стали	Термообработка	Твердость		σ_b , МПа	σ_T , МПа
			поверхности	сердцевины		
<i>HB < 350</i>	35	Н	163...192HB		550	270
	40	У	192...228HB		700	400
	45	Н	179...207HB		600	320
		У	235...262HB		780	540
		У	269...302HB		890	650
	40Х	У	235...262HB		790	640
		У	269...302HB		900	750
	40ХН	У	235...262HB		800	650
		У	269...302HB		920	750
	35ХМ	У	235...262HB		800	670
		У	269...302HB		920	790
	35Л	Н	163...207HB		550	270
	40Л	Н	147HB		520	295
45Л	У	207...235HB		680	440	
40ГЛ	У	235...262HB		850	600	
<i>HB > 350</i>	40Х	У + ТВЧ	45...50HRC	269...302HB	900	750
		А	50...59HRC	269...302HB	1000	800
	40ХН	З	48...54HRC		1600	1400
		У + ТВЧ	48...54HRC	269...302HB	920	750
	35ХМ	З	45...53HRC		1600	1400
		У + ТВЧ	48...54HRC	269...302HB	920	790
	38ХМЮА	З	45...53HRC		1700... 1950	1350... 1600
		А	57...67HRC	30...35HRC	1050	900
Обозначение термообработки: У – улучшение; Н – нормализация; З – закалка объемная; ТВЧ – закалка с нагревом токами высокой частоты; А – азотирование						

Большую часть общего количества выпускаемых в настоящее время зубчатых передач составляют одно- и двухступенчатые редукторы, поэтому в методике рассматривается двухступенчатый редуктор.

Рекомендуется следующий порядок расчета:

- выбор двигателя по требуемой номинальной мощности, заданной частоте вращения и условиям работы;
- кинематический расчет редуктора, разбивка его передаточного числа по ступеням;

- выбор материалов для шестерен и колес, определение допускаемых напряжений;

- определение крутящих моментов на шестернях и колесах ступеней;

- проектировочный расчет передачи тихоходной ступени;

- проверочные расчеты передачи тихоходной ступени;

- проектировочный расчет передачи быстроходной ступени;

- проверочные расчеты передачи быстроходной ступени.

2.3.5.2. Исходные данные для расчета

Задание на курсовой проект содержит следующие *обязательные* данные, используемые как исходные при расчете редуктора:

- кинематическая схема ММ, дающая возможно более полное представление о том, как передается вращение на ведущий вал и снимается с ведомого вала передачи;

- номинальный крутящий момент на выходном валу T_T , Нм;

- частота вращения выходного вала n_T , об/мин;

- срок службы ММ $T_{сл}$, лет;

- нагрузочная диаграмма (циклограмма), отражающая изменение крутящего момента на выходном валу в течение рабочей смены (рабочего цикла).

Кроме указанных, задание может включать в себя *дополнительные требования*, например, максимальные допустимые значения кинематической погрешности и мертвого хода, кратность максимального допустимого пикового момента, направление вращения выходного вала, максимальные допустимые габаритные размеры ММ и т. д.

2.3.5.3. Выбор двигателя

Тип двигателя ММ проектант выбирает самостоятельно, руководствуясь рекомендациями учебно-методической литературы. В настоящей методике для определенности принят наиболее широко распространенный в машиностроении трехфазный асинхронный короткозамкнутый электродвигатель.

Требуемую номинальную мощность двигателя вычисляют по формуле

$$P_{\text{тр}} = \frac{T_{\text{т}} n_{\text{т}}}{\eta_{\text{б}} \eta_{\text{т}} \eta_{\text{п}}^3}, \quad (2.27)$$

где $\eta_{\text{б}}$, $\eta_{\text{т}}$ – КПД быстроходной и тихоходной ступеней соответственно; $\eta_{\text{п}} = 0,99$ – КПД пары подшипников качения.

Как правило, в ММ применяются зубчатые передачи степеней точности не ниже 7, для которых $\eta_{\text{б}} = \eta_{\text{т}} = 0,98 \dots 0,99$.

Затем принимают типоразмер двигателя по условию

$$P_{\text{дв}} \geq P_{\text{тр}}, \quad (2.28)$$

где $P_{\text{дв}}$ – номинальная мощность двигателя по каталогу.

Допустимо превышение требуемой мощности над номинальной, если выполняется условие

$$\frac{P_{\text{тр}} - P_{\text{дв}}}{P_{\text{дв}}} \cdot 100\% \leq [\Delta P], \quad (2.29)$$

где $[\Delta P]$ – допустимая перегрузка двигателя принятого типа.

2.3.5.4. Кинематический расчет редуктора

Расчетное передаточное число редуктора

$$u_{\text{р}} = \frac{n_{\text{дв}}}{n_{\text{т}}}, \quad (2.30)$$

где $n_{\text{дв}}$ – частота вращения вала двигателя.

В том случае, если в исходных данных отсутствует требование абсолютно точного обеспечения заданной величины $n_{\text{т}}$, следует для дальнейшего проектирования принимать передаточное число редуктора u в соответствии со стандартным рядом передаточных чисел по условию

$$\frac{|u - u_{ст}|}{u_{ст}} \cdot 100\% \leq \Delta u, \quad (2.31)$$

где $u_{ст}$ – ближайшее к u_p значение передаточного числа из стандартного ряда; Δu – допустимое отклонение передаточного числа от стандартного значения.

Для цилиндрических зубчатых передач при $u_p \leq 4,5$ отклонение $\Delta u = 2,5\%$, при $u_p > 4,5$ $\Delta u = 4\%$.

В том случае, если условие (2.31) выполняется при подстановке вместо u значения u_p , можно принять либо $u = u_p$, либо $u = u_{ст}$. И то, и другое решение будет правомерно.

Для принятого u далее следует, пользуясь рекомендациями табл. 2.2, 2.3, определить передаточные числа ступеней: быстроходной $u_б$ и тихоходной $u_т$.

Таблица 2.2

Передаточные числа ступеней косозубых трехосных редукторов

$$\left(\frac{a_{вт}}{a_{вб}} = 1,4 \right)$$

Передаточное число ступени	Передаточное число редуктора u										
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5
$u_б$	3,55	4		4,5	5	5,6		6,3	7,1		8
$u_т$	2,8		3,15			3,55			4		

Таблица 2.3

Передаточные числа ступеней косозубых соосных редукторов

$$\left(\frac{\Psi_{бат}}{\Psi_{баб}} = 2 \right)$$

Передаточное число ступени	Передаточное число редуктора u										
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5
$u_б$	4,5		5	5,6		6,3	7,1		8	9	
$u_т$	2,24	2,5			2,8			3,15		3,55	

Далее следует найти частоты вращения и угловые скорости валов:

- быстроходного

$$\begin{aligned} n_{\delta} &= n_{\text{дв}}; \\ \omega_{\delta} &= \frac{\pi n_{\delta}}{30}; \end{aligned} \quad (2.32)$$

- промежуточного

$$n_{\text{п}} = \frac{n_{\delta}}{u_{\delta}}, \quad (2.33)$$

$$\omega_{\text{п}} = \frac{\omega_{\delta}}{u_{\delta}}; \quad (2.34)$$

- тихоходного

$$n_{\text{т}} = \frac{n_{\text{п}}}{u_{\text{т}}}, \quad (2.35)$$

$$\omega_{\text{т}} = \frac{\omega_{\text{п}}}{u_{\text{т}}}. \quad (2.36)$$

2.3.5.5. Материалы шестерен и колес. Допускаемые напряжения

Марки сталей и режимы термообработки для шестерен и колес редуктора назначают по рекомендациям п. 2.3.4 и данным табл. 2.1.

Допускаемое контактное напряжение при расчете на выносливость определяют по формуле

$$[\sigma_H] = \frac{\sigma_{H\text{limb}}}{S_H} K_{HL}, \quad (2.37)$$

где $\sigma_{H\text{limb}}$ – базовый предел контактной выносливости поверхности зубьев, соответствующий базовому числу циклов перемены напряжений N_{H0} ; S_H – коэффициент безопасности; K_{HL} – коэффициент долговечности.

Значения N_{H0} определяют по табл. 2.4, σ_{Hlimb} – по табл. 2.5.

Коэффициент $S_H = 1,1$ для нормализованных, улучшенных и объемно закаленных сталей (для материалов с однородной структурой); $S_H = 1,2$ для закаленных с нагревом ТВЧ и азотированных сталей (для материалов с неоднородной структурой).

Таблица 2.4

Значения N_{H0} , млн. циклов

Твердость	<i>HB</i>	200	250	300	–	–	–	–	–	–
зубьев	<i>HRC</i>	–	–	–	36	42	47	52	56	59
N_{H0}		10	12,5	25	35	50	65	85	110	150

Таблица 2.5

Значения σ_{Hlimb}

Вид термообработки	Твердость поверхности зубьев	Группа сталей	σ_{Hlimb} , МПа
Нормализация, улучшение	$HB < 350$	Углеродистые и легированные	$2HB_{cp}^* + 70$
Закалка объемная	$38...50HRC$		$18HRC_{cp}^* + 150$
Закалка с нагревом ТВЧ	$40...56HRC$		$17HRC_{cp}^* + 200$
Азотирование	$57...67HRC$	Легированные	$16HRC_{cp}^*$
Примечание. *Средние значения твердости в диапазоне (см. табл. 2.1)			

Коэффициент долговечности определяют из выражения

$$1 \leq K_{HL} = \sqrt[6]{\frac{N_{H0}}{N_{HE}}} \leq K_{HLmax}, \quad (2.38)$$

где N_{HE} – эквивалентное число циклов перемены напряжений; K_{HLmax} – максимальное допустимое значение коэффициента долговечности, зависящее от вида термообработки (при объемном упрочнении $K_{HLmax} = 2,6$; при поверхностном упрочнении $K_{HLmax} = 1,8$).

Как видно из (2.38), K_{HLmax} не может быть меньше единицы, поэтому при $N_{H0} < N_{HE}$ считают $K_{HL} = 1$.

Величина N_{HE} зависит от нагрузочной диаграммы. При постоянной нагрузке

$$N_{HE} = 60nct, \quad (2.39)$$

где n – частота вращения, об/мин, колеса (шестерни), $[\sigma_H]$ которого определяется; c – число зацеплений зуба за один оборот колеса (шестерни); t – заданный срок службы редуктора, ч.

В том случае, если задана ступенчатая нагрузочная диаграмма, N_{HE} определяют так:

$$N_{HE} = 60c \sum \left(\frac{T_i}{T_1} \right)^3 n_i t_i, \quad (2.40)$$

где T_i – крутящий момент на i -ой ступени циклограммы; T_1 – наибольший момент на циклограмме, учитываемый в расчете на выносливость; n_i, t_i – соответствующие моменту T_i частота и время работы.

Для *прямоугольной* ступени, а также для *косозубой с небольшой разностью твердости* зубьев шестерни и колеса в качестве расчетного принимают *меньшее* из двух допускаемых напряжений, определенных по материалу шестерни $[\sigma_H]_1$ и колеса $[\sigma_H]_2$.

Для *косозубой ступени с большой разностью твердости* зубьев шестерни и колеса в качестве расчетного принимают напряжение, определенное по формуле

$$[\sigma_H] = 0,45([\sigma_H]_1 + [\sigma_H]_2) \quad (2.41)$$

с проверкой по условию

$$[\sigma_H]_{\min} \leq [\sigma_H] \leq 1,25[\sigma_H]_{\min}, \quad (2.42)$$

где $[\sigma_H]_{\min}$ – меньшее из значений $[\sigma_H]_1$ и $[\sigma_H]_2$.

В том случае, если $[\sigma_H] < [\sigma_H]_{\min}$, принимают $[\sigma_H] = [\sigma_H]_{\min}$; если же $[\sigma_H] > 1,25[\sigma_H]_{\min}$, то принимают $[\sigma_H] = 1,25[\sigma_H]_{\min}$.

Допускаемое напряжение изгиба при расчете на выносливость определяют по формуле

$$[\sigma_F] = \frac{\sigma_{F\limb}}{S_F} K_{FC} K_{FL}, \quad (2.43)$$

где $\sigma_{F\limb}$ – базовый предел выносливости зубьев по излому от напряжений изгиба (см. табл. 2.6); S_F – коэффициент безопасности; K_{FC} – коэффициент влияния двухстороннего приложения нагрузки; K_{FL} – коэффициент долговечности.

Коэффициент S_F принимают в зависимости от вида термообработки из диапазона 1,7...2,2 (верхнее значение – для литых колес).

Коэффициент $K_{FC} = 1$ для односторонней нагрузки, $K_{FC} = 0,7...0,8$ для реверсивной нагрузки (бóльшие значения при $HB > 350$).

Коэффициент K_{FL} определяют по (2.43):

$$1 \leq K_{FL} = \sqrt[m]{\frac{4 \cdot 10^6}{N_{FE}}} \leq K_{FL\max}, \quad (2.44)$$

где m – показатель степени; N_{FE} – эквивалентное число циклов нагружения напряжениями изгиба; $K_{FL\max}$ – максимальное допустимое значение коэффициента долговечности.

При $HB \leq 350$, а также для шестерен и колес со шлифованными зубьями $m = 6$, $K_{FL\max} = 2,0$; при $HB > 350$, а также для шестерен и колес с нешлифованными зубьями $m = 9$, $K_{FL\max} = 1,6$.

При постоянной нагрузке значение N_{FE} находят по формуле (2.39), при изменении нагрузки по нагрузочной диаграмме – по формуле (2.44):

$$N_{FE} = 60c \sum \left(\frac{T_i}{T_1} \right)^m n_i t_i. \quad (2.45)$$

Таблица 2.6

Значения σ_{Flimb}

Вид термообработки	Твердость зубьев		Группа сталей	σ_{Flimb} , МПа
	поверхность	сердцевина		
Нормализация, улучшение	180...350HB		Углеродистые и легированные	1,8HB _{ср}
Закалка объемная	45...55HRC			550...600
Закалка с нагревом ТВЧ	40...56HRC	HB ≤ 350		1,8HB _{ср}
Азотирование	57...67HRC	24...40HRC	Легированные	43...49HRC _{ср}

Предельное допускаемое контактное напряжение для проверки ступени на прочность при перегрузках (пиковых нагрузках):

- для нормализованных, улучшенных или объемно закаленных зубьев

$$[\sigma_{H \max}] = 2,8\sigma_T; \quad (2.46)$$

- для зубьев, закаленных с нагревом ТВЧ

$$[\sigma_{H \max}] = 44 HRC_{ср}; \quad (2.47)$$

- для азотированных зубьев

$$[\sigma_{H \max}] = 35 HRC_{ср}. \quad (2.48)$$

Предельное допускаемое напряжение изгиба для проверки ступени на прочность при перегрузках определяют следующим образом:

$$[\sigma_{F \max}] = \frac{\sigma_{F \lim b}}{S_{ST}} Y_{N \max} k_{ST}, \quad (2.49)$$

где $Y_{N \max}$ – максимальное возможное значение коэффициента долговечности (для объемной термообработки $Y_{N \max} = 4,0$; для поверхностной термообработки $Y_{N \max} = 2,5$); k_{ST} – коэффициент влияния частоты приложения пиковой нагрузки (при многократном – порядка 1000 – действии перегрузок $k_{ST} = 1$); S_{ST} – коэффициент запаса прочности (обычно принимают $S_{ST} = 1,75$).

В настоящее время в преобразователях движения ММ и роботов применяются практически исключительно косозубые передачи, поэтому именно они и рассматриваются далее в методике.

2.3.5.6. Крутящие моменты на шестернях и колесах ступеней

Крутящий момент на колесе тихоходной ступени равен

$$T_{2T} = \frac{T_T}{\eta_{\Pi}}; \quad (2.50)$$

крутящий момент на шестерне тихоходной ступени и колесе быстроходной ступени

$$T_{1T} = T_{26} = \frac{T_{2T}}{u_T \eta_T \eta_{\Pi}}; \quad (2.51)$$

крутящий момент на шестерне быстроходной ступени

$$T_{16} = \frac{T_{26}}{u_6 u_T \eta_6 \eta_T \eta_{\Pi}}. \quad (2.52)$$

2.3.5.7. Проектировочный расчет тихоходной ступени

Расчетное межосевое расстояние определяют по формуле

$$a'_{вт} = 430(u_{т} \pm 1) \sqrt[3]{\frac{T_{2т} K_{H\beta т}}{u_{т}^2 \psi_{bat} [\sigma_H]_{т}^2}}, \text{ мм}, \quad (2.53)$$

где $K_{H\beta т}$ – коэффициент неравномерности распределения нагрузки по длине зуба колеса тихоходной ступени (табл. 2.8); ψ_{bat} – коэффициент ширины колеса тихоходной ступени относительно ее межосевого расстояния; $[\sigma_H]_{т}$ – допускаемое контактное напряжение для тихоходной ступени.

Знак плюс в (2.53) соответствует внешнему зацеплению, минус – внутреннему зацеплению.

Смысл коэффициента ψ_{bat} проясняет формула

$$\psi_{bat} = \frac{b_{2т}}{a_{вт}}, \quad (2.54)$$

где $b_{2т}$ – ширина венца колеса тихоходной ступени.

Стандартный ряд значений ψ_{bat} : 0,100; 0,125; 0,160; 0,200; 0,250; 0,315; 0,400; 0,500; 0,630; 0,800; 1,00; 1,25. Следует применять в расчете коэффициент ψ_{ba} из этого ряда в соответствии с рекомендациями табл. 2.7.

Далее следует принять *стандартное межосевое расстояние* $a_{вт}$, ближайшее к расчетному $a'_{вт}$, по табл. 2.9.

Определение *основных геометрических параметров* ступени целесообразно начать с выбора нормального модуля по рекомендации

$$1,50 \leq m_{нт} \leq (0,01 \dots 0,02) a_{вт}, \text{ мм}, \quad (2.55)$$

из стандартного ряда: 1,50; 1,75; 2,00; 2,25; 2,50; 2,75; 3,00; 3,50; 4,00; 4,50; 5,00; 5,50; 6,00; 7,00; 8,00...

В обоснованных случаях для передач 6 и 7 степеней точности могут быть приняты модули по ГОСТ 9563 менее 1,5 мм.

Рекомендуемые значения ψ_{ba}

Расположение колеса относительно опор	Твердость рабочих поверхностей зубьев	
	$HB_2 \leq 350$	$HB_2 > 350$
Симметричное	0,200...0,400	0,200...0,315
Несимметричное		0,200; 0,250
Консольное	0,200; 0,250	0,160; 0,200

Примечание. В шевронных передачах ψ_{ba} следует увеличить в 1,1...1,4 раза; для подвижных колес коробок передач $\psi_{ba} = 0,100...0,200$

Затем нужно предварительно задать значение угла наклона зубьев $\beta' = 8...15^\circ$ (обычно принимают $\beta' = 10^\circ$), после чего определить *расчетное суммарное число зубьев* передачи по формуле

$$z'_{ст} = \frac{2a_{вт} \cos \beta'_T}{m_{нт}}. \quad (2.56)$$

Полученное число $z'_{ст}$ округлить до ближайшего целого $z_{ст}$.

Уточненную величину угла наклона зубьев определяют по формуле

$$\beta_T = \arccos \frac{z_{ст} m_{нт}}{2a_{вт}} \quad (2.57)$$

с точностью до угловых секунд.

Расчетное число зубьев шестерни равно

$$z'_{1T} = \frac{z_{ст}}{u_T + 1}. \quad (2.58)$$

Полученное число z'_{1T} округлить до ближайшего целого z_{1T} .

Таблица 2.8

Значения коэффициента $K_{H\beta}$

Относительная ширина колеса ψ_{bd}	Шестерня расположена симметрично относительно опор		Шестерня расположена несимметрично относительно опор				Консольное расположение шестерни или колеса	
			весьма жесткий вал		менее жесткий вал			
	Твердость поверхностей зубьев HB_2							
	> 350	< 350	> 350	< 350	> 350	< 350	> 350	< 350
0,2	1,00		1,01	1,00	1,06	1,02	1,15	1,07
0,4	1,01	1,00	1,05	1,02	1,12	1,05	1,35	1,15
0,6	1,03	1,01	1,09	1,04	1,20	1,08	1,60	1,24
0,8	1,06	1,03	1,14	1,06	1,27	1,12	1,85	1,30
1,0	1,10	1,04	1,18	1,08	1,37	1,15	–	
1,2	1,13	1,05	1,25	1,10	1,50	1,18		
1,4	1,15	1,07	1,32	1,13	1,60	1,23		
1,6	1,20	1,08	1,40	1,16	–	1,28		

Примечание. Относительная ширина колеса $\psi_{bd} = 0,5(u + 1)\psi_{ba}$.

Таблица 2.9

Межосевые расстояния двухступенчатых трехосных цилиндрических редукторов

$a_{вб}$, мм	50	63	80	100	112	125	140	160	180
$a_{вт}$, мм	63	80	112	140	160	180	200	224	250
$a_{вб}$, мм	200	224	250	280	315	355	400	450	500
$a_{вт}$, мм	280	315	355	400	450	500	560	630	710

Во избежание подрезания зубьев должно выполняться условие

$$z_{1т} \geq 17 \cos^3 \beta_t. \quad (2.59)$$

Если условие (2.59) не выполняется, следует принять меньшее значение $m_{тт}$ из стандартного ряда и заново выполнить (2.56) – (2.58).

Далее находят число зубьев колеса

$$z_{2т} = z_{ст} - z_{1т} \quad (2.60)$$

и фактическое передаточное число ступени

$$u_{\text{фг}} = \frac{z_{2\text{г}}}{z_{1\text{г}}},$$

которое проверяют по условию (2.31).

По (2.19) вычисляют $d_{1\text{г}}$, $d_{2\text{г}}$, после чего находят диаметры вершин зубьев

$$\begin{aligned} d_{a1\text{г}} &= d_{1\text{г}} + 2m_{\text{нг}}; \\ d_{a2\text{г}} &= d_{2\text{г}} + 2m_{\text{нг}}. \end{aligned} \quad (2.61)$$

Ширину венца колеса предварительно определяют по выражению

$$b_{2\text{г}} = \psi_{b\text{аг}} a_{\text{вт}} \quad (2.62)$$

и окончательно принимают ближайшее значение из ряда R_{a40} номинальных линейных размеров ГОСТ 6636.

Ширину венца шестерни $b_{1\text{г}}$ принимают равной следующему за $b_{2\text{г}}$ размеру по указанному выше ряду.

2.3.5.8. Проверка тихоходной ступени на выносливость по контактными напряжениям

Действительное контактное напряжение в проектируемой передаче определяют по выражению

$$\sigma_H = \frac{6160 Z_H Z_\varepsilon}{a_{\text{вт}}} \sqrt{\frac{T_{\text{п}} (u_{\text{фг}} \pm 1)^3}{u_{\text{фг}} b_{2\text{г}}} K_{H\alpha} K_{H\beta} K_{H\nu}}, \text{ МПа}, \quad (2.63)$$

где $Z_H = 1,77 \cos \beta$ – коэффициент формы сопряженных поверхностей зубьев; Z_ε – коэффициент суммарной длины контактных линий; $K_{H\alpha}$ – коэффициент распределения нагрузки между зубьями (табл. 2.10); $K_{H\nu}$ – коэффициент динамической нагрузки (табл. 2.11).

Для определения Z_ε необходимо найти коэффициент осевого перекрытия

$$\varepsilon_{\beta} = \frac{b_{2T} \sin \beta_T}{\pi m_{nT}} \quad (2.64)$$

и проследить, чтобы выполнялось условие $\varepsilon_{\beta} \geq 0,9$, в противном случае нагрузочная способность косозубой передачи резко снизится. Возможно, что для обеспечения выполнения этого условия в геометрию передачи придется вносить изменения.

Коэффициент Z_{ε}

$$Z_{\varepsilon} = \frac{1}{\sqrt{\varepsilon_{\alpha}}}, \quad (2.65)$$

где ε_{α} – коэффициент торцевого перекрытия, определяемый по формуле (2.66):

$$\varepsilon_{\alpha} = \left[1,88 - 3,2 \left(\frac{1}{z_1} \pm \frac{1}{z_2} \right) \right] \cos \beta. \quad (2.66)$$

Знак плюс в круглых скобках соответствует внешнему зацеплению, минус – внутреннему зацеплению.

Коэффициент $K_{H\alpha}$ принимают по табл. 2.10 для степени точности, которую назначают в зависимости от окружной скорости зуба

$$v_{\pi} = \frac{\omega_{\pi} d_{1T}}{2000}, \text{ м/с}, \quad (2.67)$$

по рекомендациям: $v_{\pi} \leq 4$ м/с – степень точности 9; $4 < v_{\pi} \leq 10$ м/с – степень точности 8; $10 < v_{\pi} \leq 15$ м/с – степень точности 7; $15 < v_{\pi} \leq 30$ м/с – степень точности 6.

Таблица 2.10

Окружная скорость v , м/с	Степень точности			
	6	7	8	9
2,5	1,01	1,03	1,05	1,13
5,0	1,02	1,05	1,09	1,16
10	1,03	1,07	1,13	–
15	1,04	1,09	–	
20	1,05	1,12		
25	1,06	–		

Полученное значение контактного напряжения проверяют на соответствие условию

$$0,9[\sigma_H] \leq \sigma_H \leq 1,05[\sigma_H]. \quad (2.68)$$

В том случае, если σ_H не входит в указанные пределы, необходимо скорректировать параметры передачи. Обычно бывает достаточно изменить размеры b_{2T} и b_{1T} . Как правило, эти изменения невелики, и пересчета остальных размеров ступени не требуется.

Таблица 2.11

Степень точности	Твердость HB_2	Окружная скорость зуба v_n , м/с					
		1	2	4	6	8	10
6	≤ 350	1,01	1,02	1,03	1,04	1,06	1,07
	> 350	1,00		1,02		1,03	1,04
7	≤ 350	1,01	1,02	1,04	1,06	1,07	1,08
	> 350	1,00	1,01	1,02	1,03		1,04
8	≤ 350	1,01	1,03	1,05	1,06	1,07	1,08
	> 350	1,01		1,02	1,03	1,04	1,05
9	≤ 350	1,01	1,03	1,05	–		
	> 350	1,01		1,02	–		

2.3.5.9. Проверка тихоходной ступени на выносливость по напряжениям изгиба

Проверку выполняют по «слабому» звену зубчатой передачи, у которого меньше отношение $\frac{[\sigma_F]}{Y_F}$, где Y_F – коэффициент формы зуба, определяемый для шестерни и колеса из табл. 2.12.

Таблица 2.12

Значения коэффициента Y_F

$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F
17	4,26	21	4,01	28	3,82	40	3,70	80	3,61
18	4,20	22	4,00	30	3,80	45	3,68	100	3,60
19	4,11	24	3,92	32	3,78	50	3,65	150	
20	4,08	25	3,90	37	3,71	60	3,62	рейка	3,63

Действительное напряжение изгиба в зубе «слабого» звена проектируемой передачи равно

$$\sigma_F = 2000 Y_F Y_\beta Y_\varepsilon \frac{T_n}{b_{2T} d_{1T} m_{nT}} K_{F\alpha} K_{F\beta} K_{Fv}, \text{ МПа}, \quad (2.69)$$

где $Y_\beta = 1 - \frac{\beta}{140^\circ}$ – коэффициент наклона зубьев; Y_ε – коэффициент перекрытия зубьев, ориентировочно можно принять $Y_\varepsilon = 1$; $K_{F\alpha}$ – коэффициент распределения нагрузки между зубьями; $K_{F\beta}$ – коэффициент распределения нагрузки по ширине венца (см. табл. 2.13); K_{Fv} – коэффициент динамической нагрузки (см. табл. 2.14).

Значение $K_{F\alpha}$ определяют по формуле

$$K_{F\alpha} = \frac{4 + (\varepsilon_\alpha - 1)(n' - 5)}{4\varepsilon_\alpha}, \quad (2.70)$$

где n' – степень точности передачи.

Полученное значение σ_F не должно превышать $[\sigma_F]$ «слабого» звена более чем на 5%.

Таблица 2.13

Значения коэффициента $K_{F\beta}$

Относительная ширина колеса Ψ_{bd}	Шестерня расположена симметрично относительно опор		Шестерня расположена несимметрично относительно опор				Консольное расположение шестерни или колеса	
			весьма жесткий вал		менее жесткий вал			
	Твердость поверхностей зубьев HB_2							
	> 350	< 350	> 350	< 350	> 350	< 350	> 350	< 350
0,2	1,00		1,02	1,01	1,10	1,05	1,25	1,13
0,4	1,03	1,01	1,07	1,04	1,20	1,12	1,55	1,28
0,6	1,05	1,02	1,13	1,07	1,30	1,17	1,90	1,50
0,8	1,08	1,05	1,20	1,11	1,44	1,23	2,30	1,70
1,0	1,10	1,04	1,18	1,08	1,37	1,15	–	
1,2	1,13	1,05	1,25	1,10	1,50	1,18		
1,4	1,15	1,07	1,32	1,13	1,60	1,23		
1,6	1,20	1,08	1,40	1,16	–	1,28		

2.3.5.10. Проверка тихоходной ступени на статическую прочность при перегрузках

Максимальное контактное напряжение под действием пикового крутящего момента определяют по формуле

$$\sigma_{H \max} = \sigma_H \sqrt{\frac{P_{дв}}{P_{тр}} \left(\frac{T_{\max}}{T} \right)}, \quad (2.71)$$

где $\left(\frac{T_{\max}}{T} \right)$ – заданная кратность пикового момента.

Таблица 2.14

Значения коэффициента K_{Fv}

Степень точности	Твердость HB_2	Окружная скорость зуба v_n , м/с						
		1	2	4	6	8	10	
6	≤ 350	1,02	1,04	1,07	1,10	1,15	1,18	
	> 350	1,01	1,02	1,03	1,04	1,06	1,07	
7	≤ 350	1,03	1,06	1,11	1,16	1,22	1,27	
	> 350	1,01	1,02	1,03	1,05	1,07	1,08	
8	≤ 350	1,03	1,06	1,11	1,17	1,23	1,29	
	> 350	1,01	1,02	1,03	1,05	1,07	1,08	
9	≤ 350	1,04	1,07	1,14	–			
	> 350	1,01	1,02	1,04				

Контактная прочность при перегрузках обеспечивается, если выполняется условие: $\sigma_{H \max} \leq [\sigma_{H \max}]$.

Максимальное напряжение изгиба под действием пикового крутящего момента определяют для «слабого» звена передачи по формуле

$$\sigma_{F \max} = \sigma_F \frac{P_{\text{дв}}}{P_{\text{тр}}} \left(\frac{T_{\max}}{T} \right). \quad (2.72)$$

Изгибная прочность при перегрузках обеспечивается, если выполняется условие: $\sigma_{F \max} \leq [\sigma_{F \max}]$.

2.3.5.11. Проектировочный расчет быстроходной ступени

Межосевое расстояние $a_{\text{вб}}$ определяют в зависимости от разновидности редуктора:

- в трехосном редукторе по $a_{\text{вт}}$ и табл. 2.9;
- в соосном редукторе $a_{\text{вб}} = a_{\text{вт}}$.

Геометрические параметры $m_{\text{нб}}$; $\beta_{\text{б}}$; $d_{1\text{б}}$; $d_{2\text{б}}$; $d_{a1\text{б}}$; $d_{a2\text{б}}$; $b_{2\text{б}}$; $b_{1\text{б}}$; числа зубьев $z_{1\text{б}}$ и $z_{2\text{б}}$, а также передаточное число ступени $u_{\text{фб}}$ определяют по формулам и рекомендациям, изложенным в подпункте 2.3.5.7.

В соосном редукторе значение $b_{2\text{б}}$ находят, пользуясь рекомендацией по $\psi_{\text{баб}}$ из табл. 2.3.

Фактическое передаточное число редуктора вычисляют по выражению

$$u_{\text{ф}} = u_{\text{фб}} u_{\text{фт}} \quad (2.73)$$

и проверяют выполнение условия (2.31).

2.3.5.12. Проверочные расчеты быстроходной ступени

Проверку быстроходной ступени на выносливость по контактными и изгибными напряжениями, а также на статическую прочность при перегрузках произ-

водят по формулам и рекомендациям, изложенным в подпунктах 2.3.5.8 – 2.3.5.10.

2.4. Конические зубчатые передачи

2.4.1. Геометрические параметры конических зубчатых передач

Наибольшее распространение получили *ортогональные* конические зубчатые передачи с углом между осями шестерни и колеса 90° .

По направлению зуба различают передачи *прямозубые* (зуб расположен по образующей конуса), *с тангенциальным зубом* (зуб расположен под углом к образующей конуса) и *с круговым зубом*.

В основном применяются передачи прямозубые и с круговым зубом, так как нагрузочная способность передач с тангенциальным зубом практически не выше, чем прямозубых.

Передачи с круговым зубом по сравнению с прямозубыми имеют, при равных нагрузках, на 15...20 % меньшие габариты, работают более плавно и способны передавать вращение с большими окружными скоростями.

Недостатками передач с круговым зубом являются:

- большие величины осевых сил;
- зависимость направления осевой силы от направления вращения звена.

Эти свойства делают нежелательным применение передач с круговым зубом в преобразователях движения реверсивных ММ. Как правило, их используют *при постоянном направлении вращения* выходного звена, причем направление зуба назначают так, чтобы *осевые силы* действовали *к основаниям* образующих конусов.

На рис. 2.8 изображен фрагмент конической зубчатой передачи и показаны ее основные геометрические параметры.

Угол Σ между осями шестерни и колеса является одним из таких параметров. Ниже будут рассматриваться передачи, в которых $\Sigma = 90^\circ$.

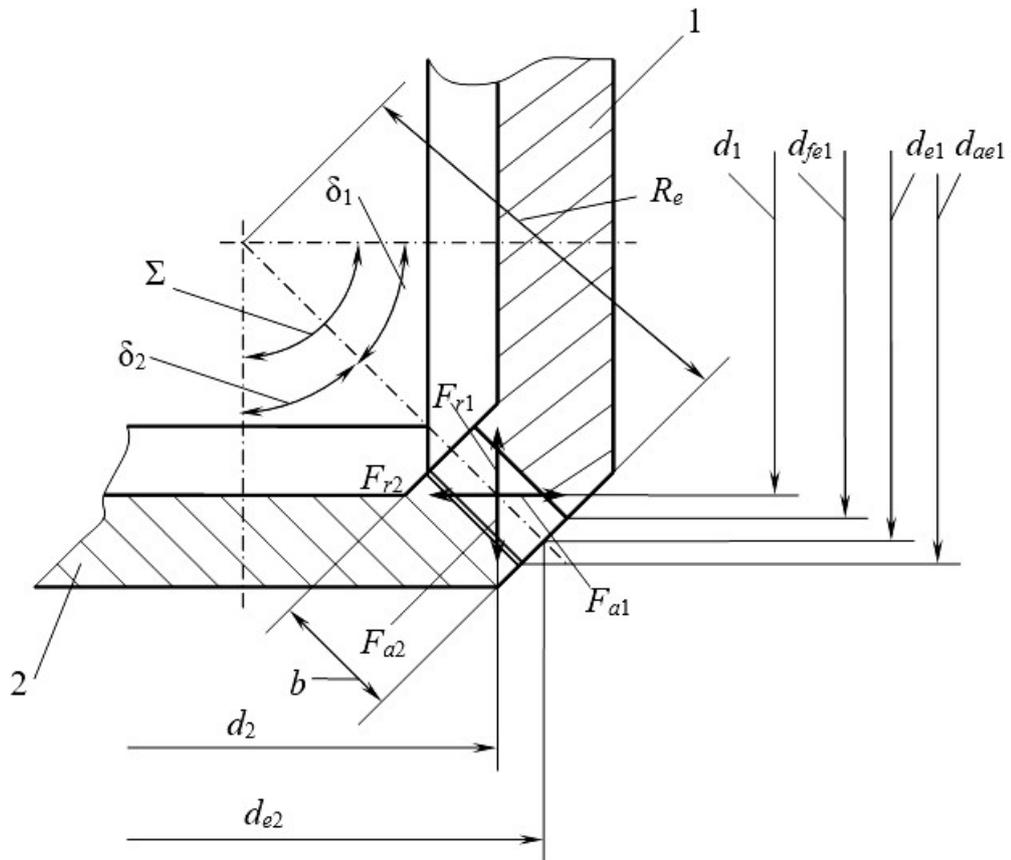


Рис. 2.8. Основные геометрические параметры конической зубчатой передачи

Внешний окружной модуль прямых зубьев обозначается m_e , круговых – m_{te} . Далее обозначения параметров передач с круговым зубом даются в скобках.

Диаметры шестерни (индекс 1) и колеса (индекс 2):

- внешние делительные

$$\begin{aligned} d_{e1} &= m_e (m_{te}) z_1; \\ d_{e2} &= m_e (m_{te}) z_2; \end{aligned} \quad (2.74)$$

- внешние окружностей вершин зубьев

$$\begin{aligned} d_{ae1} &= d_{e1} + 2h_{ae1} \cos \delta_1; \\ d_{ae2} &= d_{e2} + 2h_{ae2} \cos \delta_2; \end{aligned} \quad (2.75)$$

- внешние окружностей впадин

$$\begin{aligned}d_{fe1} &= d_{e1} - 2h_{fe1} \cos \delta_1; \\d_{fe2} &= d_{e2} - 2h_{fe2} \cos \delta_2;\end{aligned}\tag{2.76}$$

В формулах (2.74) – (2.76): z_1, z_2 – числа зубьев; h_{ae1}, h_{ae2} – высота головки зуба; h_{fe1}, h_{fe2} – высота ножки зуба; δ_1, δ_2 – углы делительных конусов.

Величины h_{ae}, h_{fe} находят с учетом коэффициентов смещения исходного контура $x_e(x_n)$ (подробнее см. учебную литературу).

Значения δ_1, δ_2 :

$$\begin{aligned}\delta_1 &= \operatorname{arctg} \frac{z_1}{z_2} = \operatorname{arctg} \frac{1}{u}; \\ \delta_2 &= 90^\circ - \delta_1.\end{aligned}\tag{2.77}$$

Внешнее конусное расстояние

$$R_e = 0,5m_e(m_{te})\sqrt{z_1^2 + z_2^2} = \frac{d_{e2}}{2 \sin \delta_2}.\tag{2.78}$$

Ширину венца колеса b вычисляют по рекомендации $b = 0,285R_e$ и принимают ближайший размер по ряду R_{a40} .

В передаче с круговым зубом к основным параметрам относится также *угол наклона зуба к образующей конуса в среднем сечении* $\beta = 35^\circ$ (рис. 2.9).

Средние делительные диаметры:

$$\begin{aligned}d_1 &\approx 0,857d_{e1}; \\ d_2 &\approx 0,857d_{e2}.\end{aligned}\tag{2.79}$$

Кроме указанных выше параметров, используют также *средний окружной модуль*, определяемый в прямозубых передачах по выражению

$$m_m = \frac{d_1}{z_1} = \frac{d_2}{z_2} \approx 0,857m_e, \quad (2.80)$$

а в передачах с круговым зубом – по выражению

$$m_{mm} \approx 0,857m_{ie} \cos \beta \approx 0,717m_{ie}, \quad (2.81)$$

и *среднее конусное расстояние*

$$R_m = R_e - 0,5b = 0,9375R_e. \quad (2.82)$$

В передачах с круговым зубом необходимо правильно ориентировать зубья на звеньях в зависимости от направления вращения. Надлежащее расположение зуба шестерни при ее вращении по часовой стрелке показано на рис. 2.9.

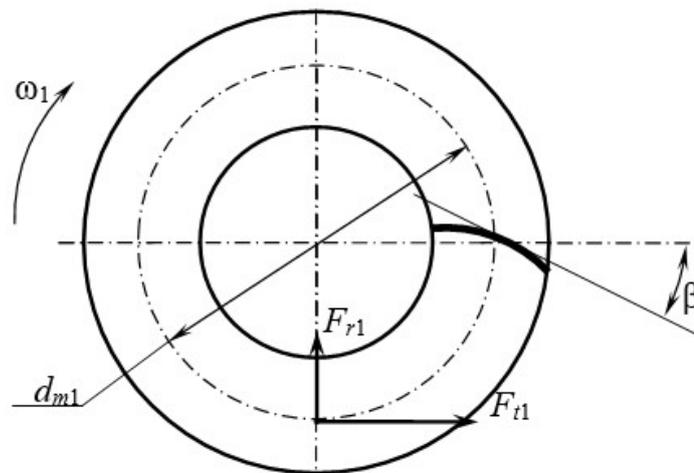


Рис. 2.9. Коническая шестерня с круговым зубом (вид со стороны вершины конуса)

2.4.2. Силы в конической зубчатой передаче

На рис. 2.8, 2.9 показаны направления сил в конической передаче.

Окружная сила (см. рис. 2.9)

$$F_t = \frac{2T_2}{d_2}. \quad (2.83)$$

В прямозубой передаче (см. рис. 2.8):

- радиальная сила на колесе, осевая на шестерне

$$F_{r2} = F_{a1} = F_t \operatorname{tg} \alpha_w \sin \delta_1 = 0,364 F_t \sin \delta_1; \quad (2.84)$$

- осевая сила на колесе, радиальная на шестерне

$$F_{a2} = F_{r1} = F_t \operatorname{tg} \alpha_w \cos \delta_1 = 0,364 F_t \cos \delta_1. \quad (2.85)$$

В передаче с круговым зубом:

- радиальная сила на колесе, осевая на шестерне

$$F_{r2} = F_{a1} = F_t (0,444 \sin \delta_1 + 0,701 \cos \delta_1); \quad (2.86)$$

- осевая сила на колесе, радиальная на шестерне

$$F_{a2} = F_{r1} = F_t (0,444 \cos \delta_1 - 0,701 \sin \delta_1). \quad (2.87)$$

2.4.3. Методика расчета конической зубчатой передачи

2.4.3.1. Общие положения

Преобразователи движения ММ, предусматриваемых заданиями на курсовое проектирование, могут содержать коническую зубчатую передачу в виде

либо одноступенчатого редуктора, либо ступени двухступенчатого редуктора. Далее рассматривается коническая передача как быстроходная ступень коническо-цилиндрического редуктора.

Общий порядок расчета – см. подпункт 2.3.5.1.

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3.

2.4.3.2. Кинематический расчет редуктора

Расчетное передаточное число редуктора определяют по (2.30) с проверкой по условию (2.31).

Распределение найденного u по ступеням необходимо выполнять с учетом следующего.

Характерной особенностью конических зубчатых передач является технологическая сложность нарезания зубьев колес при передаточных числах бóльших пяти. Поэтому в коническо-цилиндрических редукторах принимают $u_6 \leq 5$, руководствуясь рекомендациями табл. 2.15.

Таблица 2.15

Передаточные числа ступеней коническо-цилиндрических редукторов

Передаточное число ступени	Передаточное число редуктора u										
	9	10	11,2	12,5	14	16	18	20	22,4	25	≤ 28
u_6	3,15	3,55	4	4,5	5						
u_7	2,8	3,15	3,55	4	4,5	5	$\leq 5,6$				

Для конических зубчатых передач допускается отклонение $\Delta u = 4\%$ независимо от передаточного числа u_6 .

Частоты вращения и угловые скорости валов определяют по (2.32) – (2.36).

2.4.3.3. Материалы шестерен и колес. Допускаемые напряжения

Материалы звеньев конической зубчатой передачи назначают аналогично цилиндрической передаче (см подпункт 2.3.5.5).

Допускаемые напряжения передачи прямозубой конической определяют так же, как прямозубой цилиндрической; передачи с круговым зубом – как косозубой цилиндрической.

Крутящие моменты на шестернях и колесах ступеней – см. подпункт 2.3.5.6.

2.4.3.4. Расчет тихоходной ступени

Проектировочный расчет тихоходной (цилиндрической) ступени – см. подпункт 2.3.5.7.

Проверочные расчеты – см. подпункты 2.3.5.8, 2.3.5.9.

2.4.3.5. Проектировочный расчет быстроходной (конической) ступени

Внешний делительный диаметр колеса равен

$$d_{e2} = 1650 \sqrt[3]{\frac{u_{\sigma} T_{II} K_{H\beta}}{\theta_H [\sigma_H]^2}}, \text{ мм}, \quad (2.88)$$

где θ_H – коэффициент вида конических колес.

Коэффициенты $K_{H\beta}$ для передач, валы которых установлены на роликовых подшипниках, с достаточной точностью могут быть определены по аппроксимирующим формулам:

- прямозубые передачи при $HB_2 \leq 350$

$$K_{H\beta} = 0,22u_{\sigma}; \quad (2.89)$$

- прямозубые передачи при $HB_2 > 350$

$$K_{H\beta} = 0,29u_{\sigma}; \quad (2.90)$$

- передачи с круговым зубом при $HB_2 > 350$

$$K_{H\beta} = 0,24u_{\sigma}. \quad (2.91)$$

В передачах с круговым зубом при $HB_2 \leq 350$ $K_{H\beta} = 1$.

Значение $\theta_H = 1$ для прямозубых колес; θ_H для колес с круговым зубом – по следующим формулам:

- при $HB_1 \leq 350, HB_2 \leq 350$

$$\theta_H = 1,22 + 0,21u_{\sigma}; \quad (2.92)$$

- при $HB_1 > 350, HB_2 \leq 350$

$$\theta_H = 1,13 + 0,13u_{\sigma}; \quad (2.93)$$

- при $HB_1 > 350, HB_2 > 350$

$$\theta_H = 0,81 + 0,15u_{\sigma}. \quad (2.94)$$

Значение d_{e2} следует принять из ряда R_d40 ближайшее к полученному по (2.88).

Затем по (2.77) предварительно определяют углы δ_1 и δ_2 , после чего по (2.78) находят R_e .

Точность вычисления углов – до $0,0001^\circ$ (или до угловых секунд); R_e – до $0,001$ мм.

Ширину зубчатого венца находят по рекомендации, данной в пункте 2.4.1.

Определяют внешний окружной модуль

$$m_e(m_{te}) = \frac{1,4 \cdot 10^4 T_{\Pi} K_{F\beta}}{\theta_F d_{e2} b [\sigma_F]}, \text{ мм}, \quad (2.95)$$

где $K_{F\beta} = 1$ – для прямозубых передач; $K_{F\beta} = 1,08$ – для передач с круговым зубом; θ_F – коэффициент вида конических колес.

Значение $\theta_F = 0,85$ для прямозубых колес; для колес с круговым зубом θ_F определяют по следующим формулам:

- при $HB_1 \leq 350, HB_2 \leq 350$

$$\theta_F = 0,94 + 0,08u_{\sigma}; \quad (2.96)$$

- при $HB_1 > 350, HB_2 \leq 350$

$$\theta_F = 0,85 + 0,04u_{\sigma}; \quad (2.97)$$

- при $HB_1 > 350, HB_2 > 350$

$$\theta_F = 0,65 + 0,11u_{\sigma}. \quad (2.98)$$

Значение модуля до целого числа не округлять; точность – до 0,01 мм. В силовых передачах рекомендуется принимать $m_e(m_{te}) \geq 1,5$ мм.

Числа зубьев:

$$z_{2\sigma} = \frac{d_{e2}}{m_e(m_{te})}; \quad (2.99)$$
$$z_{1\sigma} = \frac{z_{2\sigma}}{u_{\sigma}}.$$

Полученные $z_{1\sigma}, z_{2\sigma}$ округлить до целых.

Фактическое передаточное число пары $u_{\phi\sigma}$ найти по (2.10) и проверить на допустимость его отклонения от стандартного.

По (2.77) с использованием $u_{\phi\sigma}$ уточнить углы δ_1 и δ_2 .

После этого следует выбрать коэффициент смещения при нарезании зубьев шестерни $x_{e1}(x_{n1})$. При $HB_1 - HB_2 \leq 100$ его принимают по табл. 2.16. В случае

отличия z_1 и u_6 от значений из табл. 2.16 $x_{e1}(x_{n1})$ принимают с округлением до табличных в *большую сторону*. При $HB_1 > 350, HB_2 > 350$ $x_{e1}(x_{n1}) = 0$, при $HB_1 - HB_2 > 100$ принимают $x_{n1} = 0$.

Коэффициент смещения при нарезании зубьев колеса $x_{e2}(x_{n2}) = -x_{e1}(x_{n1})$.

В заключение проектировочного расчета по формуле (2.75) найти d_{ae1} и d_{ae2} , а по (2.79) – d_{16} и d_{26} .

2.4.3.6. Проверка быстроходной ступени на выносливость по контактным напряжениям

Действительное контактное напряжение равно

$$\sigma_H = 470 \sqrt{\frac{F_t \sqrt{u_{\phi 6}^2 + 1}}{\theta_H d_{e2} b}} K_{H\beta} K_{Hv}, \text{ МПа.} \quad (2.100)$$

Таблица 2.16

Значения коэффициентов x_{e1}, x_{n1}

z_1	x_{e1} при передаточном числе u					x_{n1} при передаточном числе u				
	2,00	2,50	3,15	4,00	5,00	2,00	2,50	3,15	4,00	5,00
12	–	0,50	0,53	0,56	0,57	0,32	0,37	0,39	0,41	0,42
13	0,44	0,48	0,52	0,54	0,55	0,30	0,35	0,37	0,39	0,40
14	0,42	0,47	0,50	0,52	0,53	0,29	0,33	0,35	0,37	0,38
15	0,40	0,45	0,48	0,50	0,51	0,27	0,31	0,33	0,35	0,36
16	0,38	0,43	0,46	0,48	0,49	0,26	0,30	0,32	0,34	0,35
18	0,36	0,40	0,43	0,45	0,46	0,24	0,27	0,30	0,32	
20	0,34	0,37	0,40	0,42	0,43	0,22	0,26	0,28	0,29	
25	0,29	0,33	0,36	0,38	0,39	0,19	0,21	0,24	0,25	
30	0,25	0,28	0,31	0,33	0,34	0,16	0,18	0,21	0,22	
40	0,20	0,22	0,24	0,26	0,27	0,11	0,14	0,16	0,17	

Коэффициент K_{Hv} для прямозубых передач определяется по табл. 2.17, а для передач с круговым зубом – по табл. 2.11 в зависимости от степени точности, которую находят по рекомендациям для цилиндрических передач и окружной скорости

$$v_n = \frac{\omega_n d_{26}}{2000}, \text{ м/с.} \quad (2.101)$$

Таблица 2.17

Значения коэффициента K_{Hv} для прямозубых конических передач

Степень точности	Твердость HB_2	Окружная скорость зуба v_n , м/с					
		1	2	4	6	8	10
6	≤ 350	1,03	1,06	1,12	1,17	1,23	1,28
	> 350	1,02	1,04	1,07	1,10	1,15	1,18
7	≤ 350	1,04	1,07	1,14	1,21	1,29	1,36
	> 350	1,03	1,05	1,09	1,14	1,19	1,24
8	≤ 350	1,04	1,08	1,16	1,24	–	
	> 350	1,03	1,06	1,10	1,16		
9	≤ 350	1,05	1,10	–			
	> 350	1,04	1,07				

Полученное значение контактного напряжения проверяют на соответствие условию (2.68). В случае невыполнения условия следует изменить размер b . Если увеличение (или уменьшение) b на два соседних размера по ряду R_a40 не дает достаточного эффекта, необходимо перейти на другой размер d_{e2} или назначить другие материалы передачи.

2.4.3.7. Проверка быстроходной ступени на выносливость по напряжениям изгиба

Напряжения изгиба в зубьях колеса и шестерни определяют по следующим формулам:

$$\sigma_{F2} = Y_{F2} Y_{\beta} \frac{F_{t6} K_{F\beta} K_{Fv}}{\theta_F b m_e (m_{te})}; \quad (2.102)$$

$$\sigma_{F1} = \sigma_{F2} \frac{Y_{F1}}{Y_{F2}},$$

где Y_{F1} , Y_{F2} – по табл. 2.18; Y_{β} – для прямозубых передач равен единице, для передач с круговым зубом равен 0,75; K_{Fv} – для прямозубых передач – по табл. 2.19, для передач с круговым зубом – по табл. 2.14.

Полученные значения σ_{F1} , σ_{F2} не должны превышать соответствующие допускаемые напряжения более чем на 5%. При невыполнении этого условия

следует увеличить модуль и, оставив без изменения d_{e2} , пересчитать числа зубьев колеса и шестерни.

2.4.3.8. Проверочные расчеты быстроходной ступени на статическую прочность при перегрузках

Проверочные расчеты производят по формулам и рекомендациям, изложенным в подпункте 2.3.5.10.

Таблица 2.18

Значения коэффициента Y_F для конических колес

z_v	Значения $x_e(x_n)$										
	-0,5	-0,4	-0,3	-0,2	-0,1	0	0,1	0,2	0,3	0,4	0,5
12	-								3,90	3,67	3,46
14	-						4,24	4,00	3,78	3,59	3,42
17	-				4,50	4,27	4,03	3,83	3,67	3,53	3,40
20	-			4,55	4,28	4,07	3,89	3,75	3,61	3,50	3,39
25	-	4,60	4,39	4,20	4,04	3,90	3,77	3,67	3,57	3,46	3,39
30	4,60	4,32	4,15	4,05	3,90	3,80	3,70	3,62	3,55	3,47	3,40
40	4,12	4,02	3,92	3,84	3,77	3,70	3,64	3,58	3,53	3,48	3,42
50	3,97	3,88	3,81	3,76	3,70	3,65	3,61	3,57	3,53	3,49	3,44
60	3,85	3,79	3,73	3,70	3,66	3,63	3,59	3,56	3,53	3,50	3,46
80	3,73	3,70	3,68	3,65	3,62	3,61	3,58	3,56	3,54	3,52	3,50
100	3,68	3,67	3,65	3,62	3,61	3,60	3,58	3,57	3,55	3,53	3,52

Примечание. Эквивалентные числа зубьев: $z_{v1} = z_1 \cos \delta_1$; $z_{v2} = z_2 \cos \delta_2$

Таблица 2.19

Значения коэффициента K_{Fv} для прямозубых конических передач

Степень точности	Твердость $HВ_2$	Окружная скорость зуба v_n , м/с					
		1	2	4	6	8	10
6	≤ 350	1,05	1,06	1,12	1,17	1,20	1,25
	> 350	1,02	1,04	1,08	1,11	1,14	1,17
7	≤ 350	1,08	1,16	1,33	1,50	1,67	1,80
	> 350	1,03	1,05	1,09	1,13	1,17	1,22
8	≤ 350	1,10	1,20	1,38	1,58	1,78	1,96
	> 350	1,04	1,06	1,12	1,16	1,21	1,26
9	≤ 350	1,13	1,28	1,50	-		
	> 350	1,04	1,07	1,14	-		

2.5. Червячные передачи

2.5.1. Геометрические параметры червячных передач

Червячные передачи предназначены для преобразования вращательного движения между двумя *скрещивающимися осями* (межосевой угол в плане обычно равен 90°).

Различают передачи с *цилиндрическим червяком* (архимедовым, конвольютным, эвольвентным) и с *глобоидным червяком*. Более высокой нагрузочной способностью обладают передачи с глобоидным червяком, однако они значительно сложнее в изготовлении, поэтому используются только в обоснованных случаях. В основном же в ММ применяют передачи с цилиндрическим червяком (рис. 2.10), из которых, в свою очередь, наибольшее распространение получили преобразователи движения с *архимедовым червяком*. Архимедов червяк в сечении, проходящем через продольную ось, выглядит как зубчатая рейка (рис. 2.11).

Фактическое передаточное число червячной передачи (т.е. передаточное отношение от червяка к червячному колесу) равно

$$u_{\phi} = \frac{z_2}{z_1}, \quad (2.103)$$

где z_1 – *число витков червяка*.

Стандартные значения z_1 : 1; 2; 4. В обоснованных случаях в преобразователях движения ММ могут быть заданы и другие числа витков.

Обычно u лежит в пределах 8...80, но в отдельных случаях может значительно превышать верхнее значение.

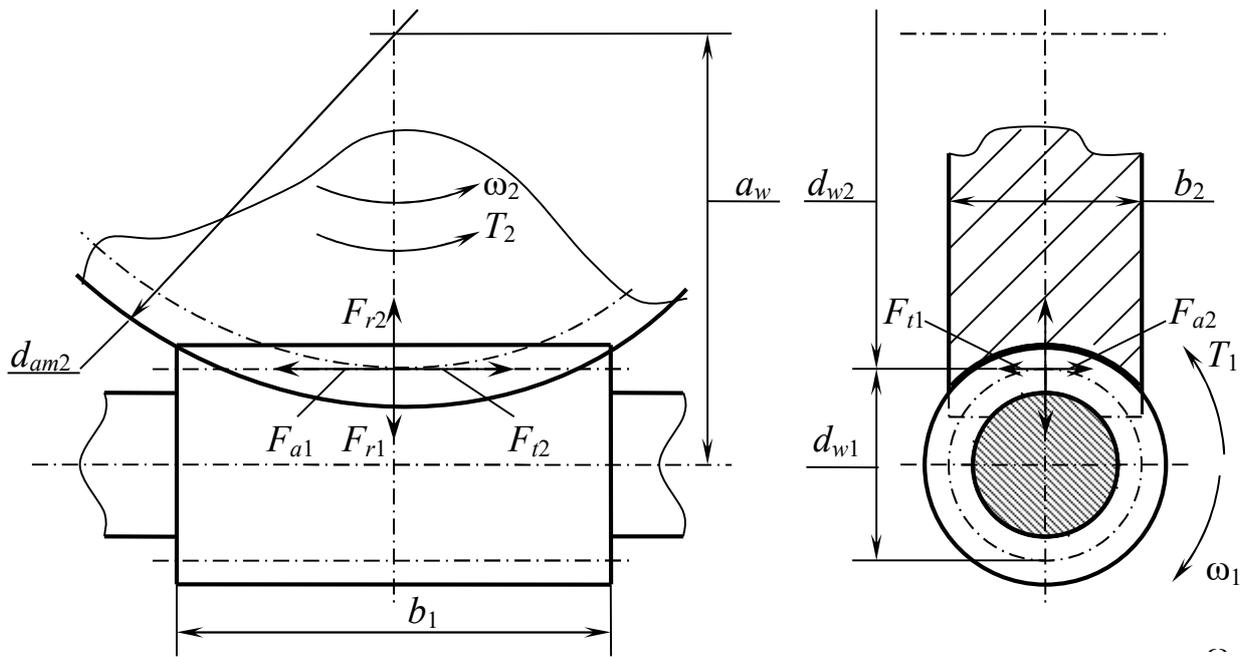


Рис. 2.10. Схема червячной передачи с цилиндрическим червяком

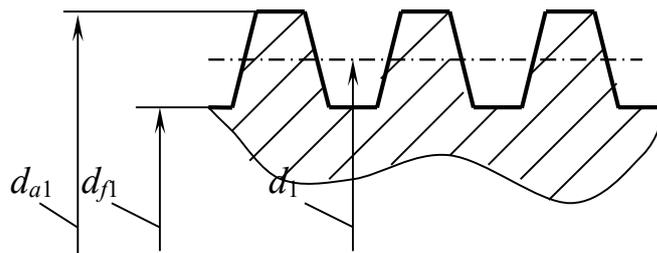


Рис. 2.11. Осевое сечение витков архимедова червяка

Диаметр делительной окружности червяка (рис. 2.11)

$$d_1 = qt, \quad (2.104)$$

где q – коэффициент диаметра, принимается из стандартного ряда.

Модуль t также стандартизован. Ряды значений q и t даны в п. 2.5.4.

Межосевое расстояние a_w (рис. 2.10) определяют по выражению

$$a_w = 0,5m(q + z_2 + 2x), \quad (2.105)$$

где x – коэффициент смещения, его величина находится в пределах: $-1 \leq x \leq 1$.

Диаметр вершин витков d_{a1} и диаметр впадин d_{f1} червяка:

$$d_{a1} = (q + 2)m; \quad (2.106)$$

$$d_{f1} = (q - 2,4)m. \quad (2.107)$$

Диаметр делительной окружности колеса

$$d_2 = mz_2;$$

диаметр вершин зубьев d_{a2} и диаметр впадин d_{f2} колеса:

$$d_{a2} = d_2 + 2(1 + x)m; \quad (2.108)$$

$$d_{f2} = d_2 - 2(1,2 - x)m; \quad (2.109)$$

наибольший диаметр колеса

$$d_{am2} = d_{a2} + \frac{6m}{z_1 + 2}. \quad (2.110)$$

Диаметр начальной окружности червяка вычисляют по формуле

$$d_{w1} = (q + 2x)m; \quad (2.111)$$

диаметр начальной окружности колеса $d_{w2} = d_2$.

Длина нарезаемой части червяка

$$b_1 = (10 + 5,5|x| + z_1)m + C, \quad (2.112)$$

где значение $C = 0$ при $x \leq 0$; при $x > 0$

$$C = \frac{100m}{z_2}. \quad (2.113)$$

Ширина венца колеса:

- при $z_1 = 1; z_1 = 2$

$$b_2 = 0,355a_w; \quad (2.114)$$

- при $z_1 = 4$

$$b_2 = 0,315a_w. \quad (2.115)$$

2.5.2. Силы в червячной передаче

Силу нормального давления на зуб колеса *в передаче с архимедовым червяком* (по аналогии с косозубой цилиндрической передачей) можно представить в виде геометрической суммы трех составляющих (рис. 2.10):

- окружной силы

$$F_{t2} = \frac{2T_2}{d_2};$$

- радиальной силы

$$F_{r2} = F_{t2} \operatorname{tg} \alpha_w = 0,364 F_{t2}; \quad (2.116)$$

- осевой силы

$$F_{a2} = F_{t1}, \quad (2.117)$$

где T_2 – крутящий момент на колесе; $\alpha_w = 20^\circ$ – угол зацепления; F_{t1} – окружная сила на червяке:

$$F_{t1} = \frac{2T_1}{d_1}; \quad (2.118)$$

T_1 – крутящий момент на червяке.

Кроме окружной, на червяк действуют радиальная $F_{r1} = F_{r2}$ и осевая $F_{a1} = F_{t2}$ силы.

КПД червячной пары определяют по формуле

$$\eta_{\text{ч}} = \frac{\text{tg}\gamma}{\text{tg}(\gamma + \varphi')}, \quad (2.119)$$

где $\gamma = \arctg \frac{z_1}{q}$ – угол подъема витка червяка; φ' – угол трения (см. п. 2.5.4).

2.5.3. Материалы червячных передач

Червяки передач малой и средней мощности, работающих с большими перерывами и редко испытывающих перегрузки, выполняют из сталей марок 40Х, 35ХМ, 40ХН. Витки таких червяков упрочняют закалкой с нагревом ТВЧ до 45...50HRC. Поверхности витков шлифуют.

Червяки тяжело нагруженных ответственных передач выполняют цементованными с закалкой до 56...63HRC с последующей шлифовкой и полировкой витков. Часто используются недорогие цементуемые стали, например, сталь 18ХГТ.

Материалы венцов червячных колес приведены в табл. 2.20. Группа материала назначается по табл. 2.21 в зависимости от скорости скольжения

$$v_s = 4 \cdot 10^{-4} n_1 \sqrt[3]{T_2}, \text{ м/с}, \quad (2.120)$$

где n_1 – частота вращения червяка, об/мин.

Материалы III группы в преобразователях движения ММ применяются редко, поэтому далее рассматриваться не будут.

Таблица 2.20

Материалы венцов червячных колес

Группа	Марка материала	Способ отливки	Механические свойства, МПа	
			σ_B	σ_T
Ia	БрО10Н1Ф1	Ц	285	165
	БрО10Ф1	К	275	200
		З	230	140
Iб	БрО5Ц5С5	К	200	90
		З	145	80
IIa	БрА10Ж4Н4	Ц	700	460
		К	650	430
	БрА10ЗМц1,5	К	550	360
		З	450	300
	БрФ9ЖЗЛ	Ц	530	245
		К	500	230
З		425	195	
IIб	ЛЦ23А6Ж3Мц2	Ц	500	330
		К	450	295
		З	400	260
III	СЧ18	З	355	-
	СЧ15	З	315	-

Примечание. Способы отливки: Ц – центробежный; К – в кокиль; З – в землю

2.5.4. Методика расчета червячной передачи

2.5.4.1. Общие положения

Преобразователи движения ММ, предусматриваемых заданиями на курсовое проектирование, могут содержать червячную передачу в виде либо одноступенчатого редуктора, либо ступени двухступенчатого редуктора. Далее рассматривается одноступенчатый червячный редуктор.

Общий порядок расчета – см. подпункт 2.3.5.1.

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3, причем требуемую номинальную мощность двигателя вычисляют по формуле

$$P_{\text{тр}} = \frac{T_2 n_2}{\eta_{\text{ч}} \eta_{\text{п}}^2}, \quad (2.121)$$

где $\eta_{\text{ч}}$ предварительно принимают равным 0,8.

2.5.4.2. Кинематический расчет передачи

Расчетное передаточное число редуктора определяют по (2.30) с проверкой по условию (2.31). В качестве $n_{\text{т}}$ используют n_2 . Допускаемое отклонение $\Delta u = 4\%$.

Таблица 2.21

Выбор группы материалов венцов червячных колес

Характеристика передачи	Скорость скольжения v_s , м/с			
	до 2	2...3	3...4	более 4
Слабо нагруженная, режим работы легкий	Ш	Пб	-	
Средне и тяжело нагруженные, режимы работы средний и тяжелый	Пб	Па	Іб	Іа
Тяжело нагруженная, режим работы весьма тяжелый	Па	Іб	Іа	

Определяют z_1 по рекомендации: при $u \geq 31,5$ $z_1 = 1$; при $u = 16...28$ $z_1 = 2$; при $u = 8...14$ $z_1 = 4$.

Определяют z_2 по формуле

$$z_2 = \frac{n_{\text{дв}}}{z_1} = \frac{n_1}{z_1} \quad (2.122)$$

с округлением до целого числа, после чего по (2.103) находят $u_{\text{ф}}$ и снова проверяют выполнение условия (2.31). В случае его невыполнения z_2 изменяют на один зуб в бóльшую или меньшую сторону.

Вычисляют частоты вращения и угловые скорости червяка и червячного колеса.

2.5.4.3. Материалы червяка и колеса. Допускаемые напряжения

По (2.127) определяют ориентировочное значение скорости скольжения, по табл. 2.21 – группу материала, а по табл. 2.20 – марку материала венца колеса.

По рекомендациям п. 2.5.3 принимают марку стали и термообработку витков червяка.

Вычисляют коэффициенты долговечности:

$$K_{HL} = \sqrt[8]{\frac{10^7}{N_{HE}}}; \quad (2.123)$$

$$K_{FL} = \sqrt[9]{\frac{10^6}{N_{FE}}}. \quad (2.124)$$

В формулах (2.123), (2.124) при $N_{HE} > 25 \cdot 10^7$ принимают $N_{HE} = 25 \cdot 10^7$, при $N_{FE} > 25 \cdot 10^7$ принимают $N_{FE} = 25 \cdot 10^7$, при $N_{FE} < 10^6$ принимают $N_{FE} = 10^6$.

По табл. 2.22 принимают коэффициент износа материала C_v , по табл. 2.23 устанавливают формулы и находят допускаемые напряжения для червячного колеса.

Таблица 2.22

Значения коэффициента износа материала

v_s , м/с	1	2	3	4	5	6	7	≥ 8
C_v	1,33	1,21	1,11	1,02	0,95	0,88	0,83	0,8

Таблица 2.23

Допускаемые напряжения для червячного колеса

Группа материалов	Червяк < 45HRC	Червяк ≥ 45 HRC	Нереверсивная передача	Реверсивная передача
	[σ] _H , Н/мм ²			
I	$K_{HL}C_v0,75\sigma_B$	$K_{HL}C_v0,9\sigma_B$	$(0,08\sigma_B+0,25\sigma_T)K_{FL}$	$0,16\sigma_BK_{FL}$
II	$250 - 25v_s$	$300 - 25v_s$		

2.5.4.4. Проектировочный расчет передачи

Межосевое расстояние определяют по формуле

$$a_w = 610 \sqrt[3]{\frac{T_2}{[\sigma_H]^2}}. \quad (2.125)$$

где T_2 – в ньютонметрах.

Полученное значение a_w округляют до ближайшего по ряду $R_a 20$.

Определяют модуль зацепления

$$m = (1,5 \dots 1,7) \frac{a_w}{z_2}, \text{ мм}, \quad (2.126)$$

и принимают ближайшее стандартное значение из первого ряда: 2,50; 3,15; 4,00; 5,00; 6,30; 8,00; 10,00; 12,50; 16,00. Допускается также использование второго ряда: 3,00; 3,50; 6,00; 7,00; 12,00.

Коэффициент диаметра определяют по рекомендации

$$q = (0,212 \dots 0,250) z_2. \quad (2.127)$$

Полученную величину округляют до стандартной из первого ряда: 6,3; 8,0; 10,0; 12,5; 16,0 или из второго ряда: 7,1; 9,0; 11,2; 14,0; 18,0. По ГОСТ 19672-74 допускается также применение значений q : 7,5 и 12,0.

Коэффициент смещения нарезающего инструмента находят из выражения

$$x = \frac{a_w}{m} - 0,5(q + z_2) \quad (2.128)$$

с проверкой на соответствие критерию из п. 5.2.1.

По (2.105) определяют фактическое межосевое расстояние.

Вычисляют основные геометрические размеры звеньев передачи по формулам (2.106) – (2.115).

Кроме того, находят делительный угол подъема витков

$$\gamma = \arctg \frac{z_1}{q}. \quad (2.129)$$

2.5.4.5. Проверочные расчеты

Уточняют КПД передачи по формуле (2.119), причем угол трения φ' принимают по табл. 2.24 в зависимости от фактической скорости скольжения

$$v_{s\varphi} = \frac{u_{\varphi} \omega_2 d_1}{2000 \cos \gamma}, \text{ м/с.} \quad (2.130)$$

Таблица 2.24

Значения угла трения

$v_{s\varphi}$	0,1	0,5	1	1,5	2	2,5	3	4	7	10	15
φ'	4°30'... ...5°10'	3°10'... ...3°40'	2°30'... ...3°10'	2°20'... ...2°50'	2°00'... ...2°30'	1°40'... ...2°20'	1°30'... ...2°00'	1°20'... ...1°40'	1°00'... ...1°30'	0°55'... ...1°20'	0°50'... ...1°10'
Примечание. Меньшие значения φ' – для материалов группы I, бóльшие – для материалов групп II и III.											

В том случае, если получается $\eta_{\varphi} < 0,8$, следует заново подсчитать требуемую мощность электродвигателя и при необходимости назначить двигатель более мощный.

Уточняют значение $[\sigma]_H$ по фактической скорости скольжения и формулам из табл. 2.23.

Контактное напряжение в зацеплении сравнивают с допусковым:

$$\sigma_H = 340 \sqrt{\frac{F_{t2} K}{d_1 d_2}} \leq [\sigma]_H, \quad (2.131)$$

где K – коэффициент нагрузки (принимается в зависимости от окружной скорости колеса (формула (2.139)): при $v_2 \leq 3$ м/с $K = 1$; при $v_2 > 3$ м/с $K = 1,1 \dots 1,3$.

Скорость v_2 определяют по формуле

$$v_2 = \frac{\omega_2 d_2}{2000}, \text{ м/с.} \quad (2.132)$$

Допускаемая недогрузка передачи 15 %, допускаемая перегрузка 5 %.

Напряжение изгиба в зубе колеса сравнивают с допускаемым:

$$\sigma_F = 0,7 \frac{Y_{F2} F_{t2} K}{b_2 m} \leq [\sigma]_F, \quad (2.133)$$

где Y_{F2} – коэффициент формы зуба колеса, принимается по табл. 2.25 в зависимости от эквивалентного числа зубьев $z_{v2} = z_2 / \cos^3 \gamma$.

Таблица 2.25

Коэффициент формы зуба червячного колеса

z_{v2}	Y_{F2}	z_{v2}	Y_{F2}	z_{v2}	Y_{F2}	z_{v2}	Y_{F2}
20	1,98	30	1,76	40	1,55	80	1,34
24	1,88	32	1,71	45	1,48	100	1,30
26	1,85	35	1,64	50	1,45	150	1,27
28	1,80	37	1,61	60	1,40	300	1,24

Как правило, получается $\sigma_F \ll [\sigma]_F$, так как нагрузочная способность червячных передач ограничивается не изгибной, а контактной выносливостью.

Выполняют проверку (тепловой расчет) редуктора на нагрев. Цель проверки – определить температуру масла в редукторе, которая не должна превышать допускаемую $[t]_M = 80 \dots 95$ °С.

Температуру масла в корпусе червячной передачи при непрерывной работе находят по формуле

$$t_m = t_B + \frac{P_1(1 - \eta_\phi)}{K_t A}, \quad (2.134)$$

где t_B – температура окружающего воздуха, принимают $t_B = 20$ °С; P_1 – мощность на червяке, Вт; K_t – коэффициент теплопередачи (среднее значение коэффициента $K_t = 13$ Вт/(м²·град)); A – площадь теплоотдающей поверхности корпуса (табл. 2.26), м².

Таблица 2.26

Площадь теплоотдающей поверхности корпуса червячного редуктора, м²

Межосевое расстояние, мм	80	100	125	140	160	180	200	224
Площадь A	0,19	0,24	0,36	0,43	0,56	0,67	0,8	1

При невыполнении условия $t_m \leq [t]_m$ следует увеличить с помощью ребрения площадь поверхности теплоотдачи. Если этой меры недостаточно, то необходимо предусмотреть специальные средства охлаждения (обдув корпуса вентилятором, введение в конструкцию холодильника для масла).

2.6. Планетарные зубчатые передачи

2.6.1. Общие сведения о планетарных передачах

Конструкции планетарных передач весьма многообразны. Простейшая планетарная передача (рис. 2.12) состоит из центрального *солнечного* зубчатого колеса 1 с наружными зубьями, центрального *корончатого* зубчатого колеса 3 с внутренними зубьями, *сателлитов* 2 с внешними зубьями, которые входят в зацепление одновременно с солнечным и корончатым колесами, и *водила* H , на котором расположены оси сателлитов.

При закреплённом корончатом колесе 3 ($\omega_3 = 0$) вращение солнечного колеса с угловой скоростью ω_1 вызывает вращение сателлитов относительно собственных осей с угловой скоростью ω_2 , что вызывает качение сателлитов по

корончатому колесу и приводит к их перемещению по круговой орбите радиуса R_H , а следовательно, и водила H с угловой скоростью ω_H .

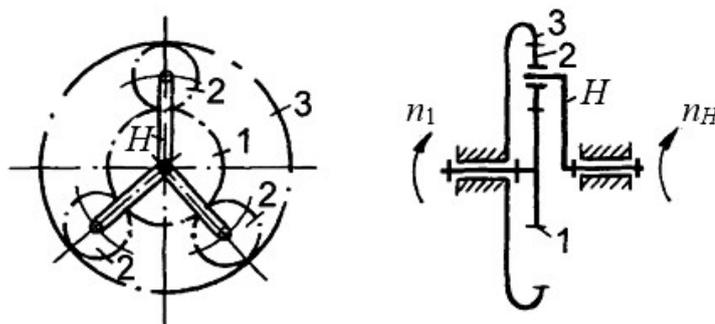


Рис. 2.12. Планетарная передача схемы $2K-H$

Сателлиты вращаются относительно водила со скоростью $\omega_2^H = \omega_2 - \omega_H$ и вместе с водилом совершают переносное движение. Их движения напоминают движения планет, поэтому передачу и называют планетарной.

Основными звеньями планетарной передачи являются те, которые воспринимают внешние моменты. Такими звеньями являются солнечные и корончатые колеса, т. е. два центральных колеса ($2K$) и водило H . Сокращенное обозначение такого планетарного механизма $2K-H$.

Планетарный механизм $2K-H$ наиболее часто используется в ММ, так как имеет высокий КПД и технологичную конструкцию. Поэтому ниже будет дана методика расчета именно этой схемы.

Подробно и полно планетарные передачи рассмотрены в специальной литературе. Здесь же следует сказать о весьма перспективной конструкции – так называемом эксцентриково-планетарном механизме, вариант которого – *механизм с параллельными кривошипами* – приведен на рис. 2.13.

Планетарный механизм $K-H-V$ имеет три основных звена: центральное колесо 2, водило H и вал V . Водило представляет собой эксцентрик, на котором установлен сателлит 1. Сателлит передает вращение на вал V кривошипами 3 – эксцентриками, эксцентриситет которых равен эксцентриситету водила.

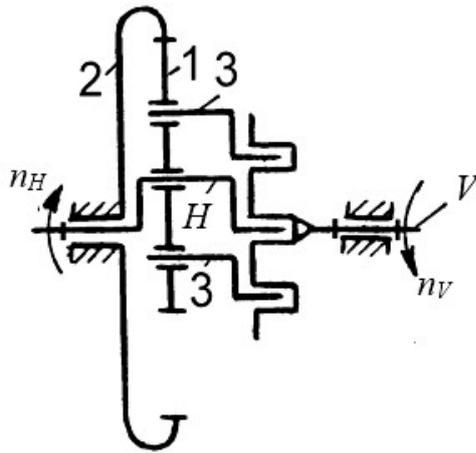


Рис. 2.13. Планетарный механизм с параллельными кривошипами (схема $K-H-V$)

За один оборот водила сателлит, совершая сложное плоское движение, поворачивается относительно своей оси на угол, соответствующий разности чисел зубьев центрального колеса и сателлита, в направлении, противоположном направлению вращения водила, т. е. передаточное число такого преобразователя движения равно

$$u_{H-V}^{(2)} = -\frac{z_2}{z_2 - z_1}. \quad (2.135)$$

Для предотвращения заклинивания зубьев в зацеплении должно выполняться условие

$$z_2 - z_1 \geq \begin{cases} 8 & \text{при } h_a^* = 1; \\ 7 & \text{при } h_a^* = 0,8, \end{cases} \quad (2.136)$$

где h_a^* – высота головки зуба.

Очевидно, что при достаточно больших числах зубьев звеньев этой пары одна ступень передачи $K-H-V$ может по передаточному числу заменить двухступенчатый цилиндрический редуктор.

Дальнейшие исследования в направлении уменьшения размеров и повышения технологичности изготовления передачи привели к появлению так называемого *планетарно-цевочного редуктора* (рис. 2.14), в котором вместо эвольвентного зацепления применяют цевочное.

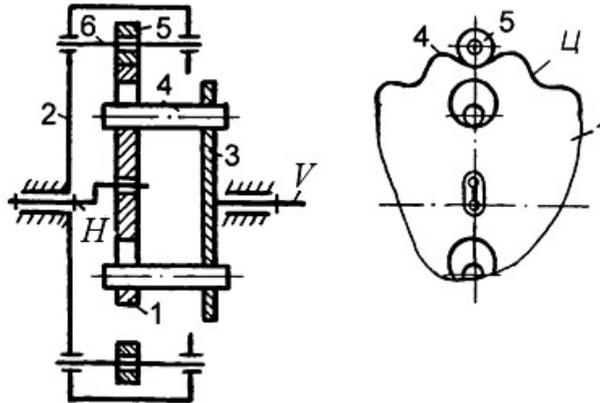


Рис. 2.14. Планетарно-цевочный редуктор схемы *K-H-V*

Сателлит 1 имеет зубья, выполненные по циклоиде (Ц), и смонтирован на водиле *H*. Центральное колесо 2 содержит оси 6 с установленными на них с возможностью вращения цевками 5, взаимодействующими с зубьями сателлита. Выходной вал *V* соединен с диском 3, в отверстия которого с натягом вставлены пальцы 4, входящие в отверстия сателлита. Пальцы, контактируя со стенками отверстий, выполняют функцию параллельных кривошипов в схеме по рис. 2.13.

Далее будет рассмотрена планетарная передача, показанная на рис. 2.12.

2.6.2. Передаточное число и условия существования планетарного механизма

Передаточное число передачи *2K-H* равно

$$u_{1H}^{(3)} = 1 + \frac{z_3}{z_1}, \quad (2.137)$$

где $u_{1H}^{(3)}$ – передаточное число от солнечного колеса к водилу при неподвижном солнечном колесе.

При проектировании следует учитывать, что передаточное число данного механизма не должно превышать 8, в крайнем случае – 9. При необходимости обеспечения большего передаточного числа преобразователь движения выполняют многоступенчатым, причем каждая ступень представляет собой передачу по рис. 2.12.

Числа зубьев колес выбирают так, чтобы отсутствовали подрезание и заклинивание зубьев. Для этого число зубьев солнечного колеса должно удовлетворять следующим рекомендациям: $z_1 \geq 24$ при выполнении солнечного колеса из стали нормализованной и улучшенной твердостью $HB \leq 350$; $z_1 \geq 21$ – из стали закаленной ТВЧ твердостью $HRC \leq 52$; $z_1 \geq 18$ – из стали цементованной твердостью $HRC \geq 52$. Минимальное число зубьев корончатого колеса

$$z_{2\min} \geq \begin{cases} 85 \text{ при } h_a^* = 1; \\ 58 \text{ при } h_a^* = 0,8. \end{cases} \quad (2.138)$$

Для существования планетарного механизма необходимо, чтобы числа зубьев соответствовали условиям сборки, соосности и соседства сателлитов.

Из условия соосности следует, что число зубьев сателлита должно соответствовать равенству

$$z_2 = \frac{z_3 - z_1}{2}. \quad (2.139)$$

По условию сборки необходимо, чтобы выполнялось равенство

$$\frac{z_3 + z_1}{C} = \gamma, \quad (2.140)$$

где C – число сателлитов (обычно $C = 3$); γ – целое число.

Условие соседства сателлитов выглядит следующим образом:

$$z_2 + 2 \leq (z_1 + z_2) \sin \frac{\pi}{C}. \quad (2.141)$$

Следует также иметь в виду, что числа зубьев колес z_1 и z_3 должны быть или *оба четные*, или *оба нечетные*.

2.6.3. Материалы планетарных передач

Для изготовления звеньев планетарных передач используют те же марки сталей и методы термообработки, что и для цилиндрических зубчатых передач, с учетом следующих рекомендаций.

Так как зуб солнечного колеса более часто входит в зацепление, чем зуб сателлита, то при твердости зубьев сателлита $HB \leq 350$ твердость зубьев солнечного колеса назначают на 50...70 единиц выше, чем сателлита. При твердости зубьев сателлита $HB > 350$ твердости солнечного колеса и сателлита назначают одинаковыми.

Для изготовления водил используют углеродистые стали. Корпуса планетарных передач ММ обычно изготавливают из легких материалов и сплавов.

2.6.4. Методика расчета планетарной передачи

2.6.4.1. Общие положения

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3.

2.6.4.2. Кинематический расчет передачи

Цель кинематического расчета – определение чисел зубьев солнечного и корончатого колес и сателлита, обеспечивающих требуемое передаточное число, а также число сателлитов.

Находят расчетное передаточное число

$$u_{1Hp}^{(3)} = \frac{n_1}{n_H}, \quad (2.142)$$

где n_H – заданная частота вращения водила.

Определение чисел зубьев колес планетарных передач производят обычно методом подбора, задаваясь числом зубьев солнечного колеса по рекомендациям п. 2.6.2 и обеспечивая при этом правильность зацепления.

Задавшись z_1 , предварительно находят z_3 по формуле

$$z_3 = z_1(u_{1Hp}^{(3)} - 1). \quad (2.143)$$

Затем из условия соосности вычисляют число зубьев сателлита z_2 и проверяют выполнение условий сборки и соседства сателлитов. При невыполнении какого-либо из условий изменяют числа зубьев колес на $\pm 1...3$ зуба и добиваются выполнения условия.

После определения чисел зубьев находят фактическое передаточное число

$$u_{\phi} = 1 + \frac{z_3}{z_1}. \quad (2.144)$$

Вычисляют отклонение фактического передаточного числа от расчетного. Допускаемое отклонение передаточного отношения обычно принимают равным 4 %.

При невыполнении этого условия необходимо число z_1 уменьшить, снова найти числа z_2 и z_3 , после чего провести проверку механизма по условиям соосности, сборки и соседства сателлитов.

2.6.4.3. Материалы и допускаемые напряжения

Материалы звеньев передачи назначают с учетом рекомендаций, приведенных в п. 2.6.3.

Допускаемые напряжения в парах центральное колесо – сателлит и сателлит – корончатое колесо определяют так же, как в цилиндрических зубчатых передачах.

2.6.4.4. Проектировочный расчет передачи

Межосевое расстояние пары солнечное колесо – сателлит

$$a_w = K_a (u + 1) \sqrt[3]{\frac{T_1 K_{H\beta} K_C}{[\sigma]_H^2 u^2 C \psi_{ba}}}, \text{ мм}, \quad (2.145)$$

где u – передаточное число пары солнечное колесо – сателлит (отношение чисел зубьев большего колеса к числу зубьев меньшего); T_1 – вращающий момент на солнечном колесе, Нм; K_C – коэффициент неравномерности распределения нагрузки между сателлитами (при наличии механизма выравнивания нагрузки $K_C = 1,1 \dots 1,2$; при отсутствии – $K_C = 1,5 \dots 2,0$); C – число сателлитов; ψ_{ba} – коэффициент ширины зубчатого венца сателлита.

Коэффициент ψ_{ba} :

$$\psi_{ba} = \frac{b_C}{a_w}, \quad (2.146)$$

где b_C – ширина венца сателлита.

Значение ψ_{ba} принимают из ряда стандартных значений: 0,100; 0,105; 0,160; 0,200; 0,250; 0,315; 0,400; 0,500; 0,630; 0,800 (желательно принимать $\psi_{ba} = 0,400$ для материалов колес твердостью HB ≤ 350 ; $\psi_{ba} = 0,315$ при твердости HRC ≤ 50 ; $\psi_{ba} = 0,250$ при твердости HRC > 50).

Затем находят делительный диаметр меньшего из колес пары солнечное колесо – сателлит. Пусть меньшее – сателлит, тогда

$$d_C = \frac{2a_w}{u + 1}. \quad (2.147)$$

Определяют модуль зубьев

$$m = \frac{d_1}{z_1} = \frac{d_c}{z_c}. \quad (2.148)$$

Полученное значение модуля округляют до стандартного, уточняют значения делительных диаметров солнечного колеса, сателлита, корончатого колеса, а также межосевого расстояния.

Затем по приведенным выше формулам вычисляют диаметры окружностей вершин и впадин зубьев всех колес передачи.

Определяют окружную скорость солнечного колеса и по ее величине назначают степень точности планетарной передачи по аналогии с цилиндрической зубчатой передачей.

Для ММ рекомендуют 6 или 7 степени точности планетарных передач.

2.6.4.5. Проверочные расчеты

Проверку зубьев в паре солнечное колесо – сателлит на выносливость по контактным напряжениям производят в соответствии с формулой

$$\sigma_H = 22,4 Z_H Z_M Z_\varepsilon \sqrt{\frac{T_1 (u+1)^3 K_{H\alpha} K_{H\beta} K_{H\nu}}{b_c u C}} \leq [\sigma]_H, \quad (2.149)$$

где коэффициенты Z_M , Z_H , Z_ε определяют так же, как при расчете цилиндрической зубчатой передачи.

При невыполнении условия (2.149) в расчет следует ввести поправку путем изменения ширины венца сателлита.

Проверку зубьев пары солнечное колесо – сателлит на выносливость по напряжениям изгиба производят для того колеса, чье отношение $\frac{[\sigma]_F}{Y_F}$ меньше, по аналогии с цилиндрическими зубчатыми передачами.

Допустимые отклонения фактических напряжений от допускаемых те же, что и для цилиндрических зубчатых передач.

Проверочный расчет зубьев колес планетарной передачи при перегрузках аналогичен такому же расчету цилиндрической зубчатой передачи.

2.7. Волновые зубчатые передачи

2.7.1. Общие сведения

Конструкция волновой зубчатой передачи поясняется рис. 2.15. По своей сути она представляет собой планетарную передачу, у которой одно из колес выполнено в виде *гибкого венца*.

Волновая зубчатая передача состоит из *гибкого колеса 1* с наружными зубьями, *жесткого колеса 2* с внутренними зубьями и *генераторов волн 3*.

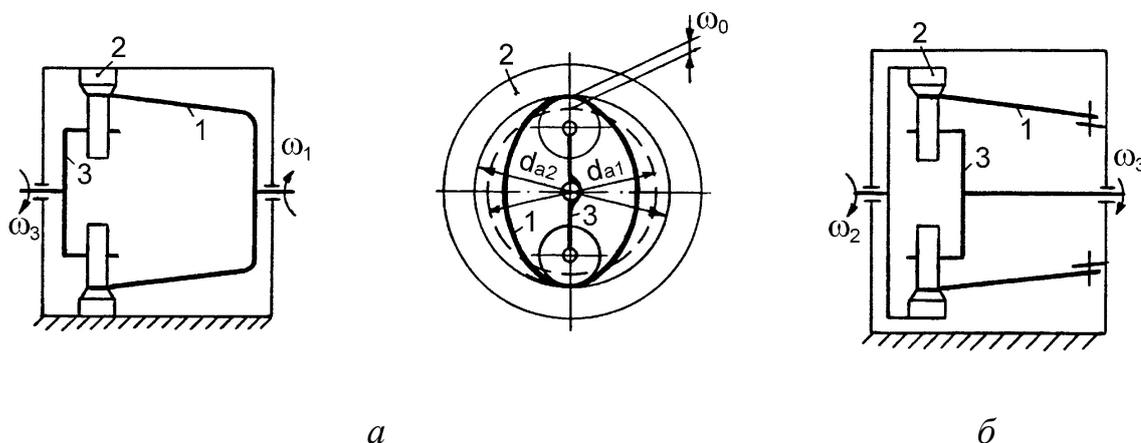


Рис. 2.15. Схема волновой зубчатой передачи

Наружный диаметр d_{a1} гибкого колеса в недеформированном состоянии меньше внутреннего диаметра d_{a2} жесткого колеса на величину

$$d_{a2} - d_{a1} = 2\omega_0, \quad (2.150)$$

где ω_0 – размер, показанный на рис. 2.15.

Возможны исполнения передачи с ведомым гибким колесом (рис. 2.15, а) и с ведомым жестким колесом (рис. 2.15, б).

О конструкции гибкого колеса будет сказано ниже.

Используются генераторы волн двух типов – кулачковый и дисковый.

Кулачковый генератор волн (рис. 2.16) содержит кулачок 1, рабочая поверхность которого спрофилирована по эллипсу, и размещенный на нем гибкий шариковый подшипник 2.

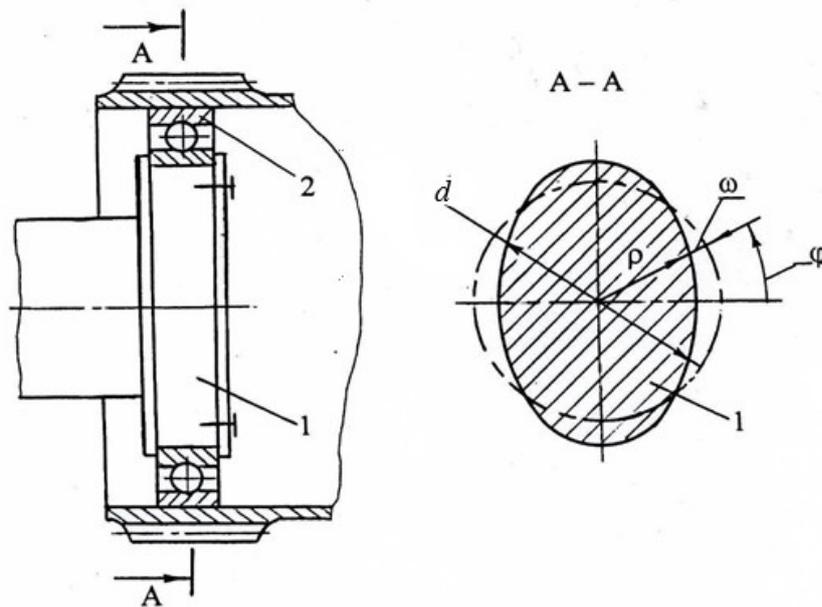


Рис. 2.16. Кулачковый генератор волн

Параметры гибких подшипников указаны в табл. 2.27.

Дисковый генератор волн (рис. 2.17) содержит диски 1 и 2, расположенные относительно быстроходного вала 3 с эксцентриситетом e посредством шариковых подшипников.

Диапазон передаточных чисел волновых зубчатых передач: $80 \leq u \leq 400$.

Вследствие многопарности зацепления для этих передач характерна весьма высокая кинематическая точность. При сравнительно малых габаритах и массе волновые передачи способны создавать на выходном звене вращаю-

щие моменты до 150 кНм. На волновом принципе основаны механизмы для передачи вращения в герметично закрытое пространство.

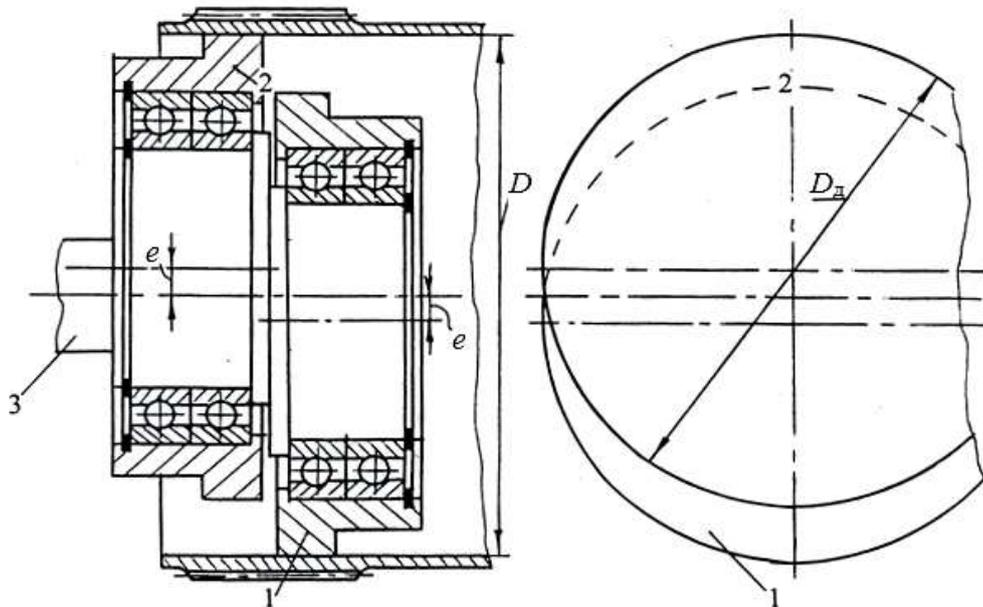


Рис. 2.17. Дискосый генератор волн

Недостатки волновых передач:

- сложность изготовления;
- пониженный по сравнению с цилиндрическими зубчатыми передачами КПД ($\eta_{вп} = 0,8 \dots 0,9$);
- ограниченность частоты вращения кулачкового генератора волн максимальной частотой вращения гибкого подшипника;
- высокая чувствительность к качеству смазки.

2.7.2. Передаточное число волновой передачи

При неподвижном жёстком зубчатом колесе 2 (рис. 2.15, а) передаточное число от вала генератора волн 3 к валу гибкого колеса 1:

$$u_{31}^{(2)} = -\frac{n_6}{n_r} = -\frac{\omega_3}{\omega_1} = -\frac{z_1}{z_2 - z_1} = -\frac{z_1}{K_z v} = -\frac{d_1}{d_2 - d_1}. \quad (2.151)$$

Знак минус указывает на разные направления вращения ведущего и ведомого звеньев.

Передаточное число от вала генератора волн 3 к валу жесткого колеса 2 при неподвижном гибком колесе 1 (рис. 2.15, б) вычисляются по формуле

$$u_{32}^{(1)} = \frac{n_6}{n_T} = \frac{\omega_3}{\omega_2} = \frac{z_2}{z_2 - z_1} = \frac{z_2}{K_z v} = \frac{d_2}{d_2 - d_1}. \quad (2.152)$$

В формулах (2.151), (2.152): $\omega_1, \omega_2, \omega_3$ – угловая скорость соответственно гибкого колеса, жесткого колеса и генератора волн; z_1, z_2 – числа зубьев гибкого и жесткого колес; d_1, d_2 – диаметры делительных окружностей гибкого и жесткого колес; v – число волн деформации ($v = 1, 2, 3 \dots$, обычно $v = 2$, реже 3); K_z – коэффициент кратности (обычно $K_z = 1$; при $u < 70$ $K_z = 2$; при $u < 45$ $K_z = 3$).

2.7.3. Геометрические параметры волновой передачи

Основные геометрические параметры волновой передачи показаны на рис. 2.18 (буквами *a, б, в* на рис. 2.18 обозначены варианты конструктивного исполнения гибкого колеса).

Параметры гибкого колеса:

- внутренний диаметр D ;
- делительный диаметр зубчатого венца d_1 ;
- ширина зубчатого венца гибкого колеса b_1 ;
- толщина зубчатого венца гибкого колеса h_1 ;
- толщина оболочки гибкого колеса $h_2 = h_3$;
- длина гибкого колеса L ;
- конструктивные размеры – $b_3; h_4; h_5$.

Параметры жёсткого колеса:

- делительный диаметр зубчатого венца d_2 ;
- ширина зубчатого венца гибкого колеса b_2 ;
- конструктивные размеры – $c; h_6$.

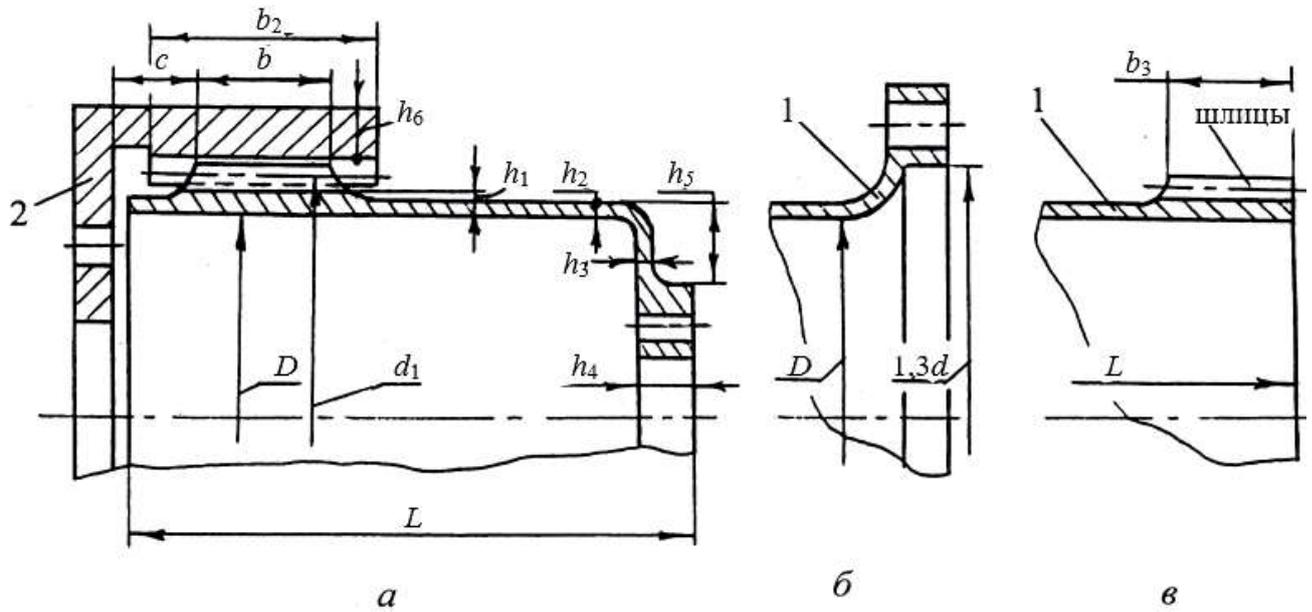


Рис. 2.18. Геометрические параметры волновой передачи

Таблица 2.27

Подшипники шариковые однорядные для волновых передач (ГОСТ23179-78)

Обозначение подшипника	Размеры, мм					Число шариков $z_{ш}$	n_{max} , об/мин	
	d	D	B	r	$d_{ш}$			
806	30	42	7	0,5	3,969	21	3000	
808	40	52	8			23		
809	45	62	9			5,953		21
812	60	80	13			7,144		23
815	75	100	15	1,0	9,128	21	1500	
818	90	120	18			11,113		23
822	110	150	24			14,288		21
824	120	160				19,050		23
830	150	200	30	22,225				
836	180	240	35	28,575	23			
844	220	300	45					
848	240	320	48	2,5	36,513	23		
860	300	400	60					
862	310	420	70	3,5	44,450	23		
872	360	480	72					

П р и м е ч а н и е. d – внутренний диаметр подшипника; D – наружный диаметр подшипника; B – ширина подшипника; r – радиус скругления кромок колец; $d_{ш}$ – диаметр шариков.

Важным геометрическим параметром зацепления является модуль m , величина которого оговаривается стандартным рядом, мм: 0,20; 0,22; 0,28; 0,30; 0,35; 0,40; 0,45. Ряд содержит также значения меньше 0,20 мм, однако их применяют редко из-за сложности нарезания внутренних зубьев жесткого колеса.

2.7.4. Материалы волновых передач

В качестве материалов зубчатых колес, как правило, используют стали, которые после термообработки имеют твердость 220...330 *HV* (табл. 2.28). В отдельных случаях колеса изготавливают из бериллиевой бронзы БрБ2 или пластмассы.

Таблица 2.28

Марки сталей для изготовления зубчатых колес волновых передач

Сталь	Твердость, <i>HV</i>	Предел выносливости σ_{-1} , МПа
40Х	280...300	500
40ХНМА	310...330	550
30ХГСА	300...320	530
38ХНЗВА	310...330	550
38ХМЮА	220...240	550
ШХ15	260...280	420
Х18Н10Т	220...240	350

2.7.5. Методика расчета волновой передачи

2.7.5.1. Общие положения

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3.

2.7.5.2. Кинематический расчет передачи

Кинематический расчет волновой передачи заключается в подборе чисел зубьев гибкого и жесткого колес по принятому в соответствии с исходными данными передаточным числом (формулы (2.152), (2.153)).

Число зубьев z_1 гибкого колеса определяют в зависимости от того, какое колесо вращается. При подвижном гибком колесе

$$z_1 = uK_z v, \quad (2.153)$$

при подвижном жестком колесе

$$z_1 = (u - 1)K_z v. \quad (2.154)$$

2.7.5.3. Материалы и допускаемое напряжение

Материал зубчатых колес принимают в соответствии с п. 2.7.4.

Допускаемые напряжения смятия на боковых поверхностях зубьев определяют в виде:

$$[\sigma]_{\text{см}} = 15,7 K_u K_n K_{d1}, \text{ МПа}, \quad (2.155)$$

где $K_u = \frac{u - 20}{u}$ – коэффициент передаточного числа; $K_n = \frac{10}{\sqrt[3]{n_6}}$ – коэффициент

частоты вращения генератора волн (в понижающей передаче частота вращения генератора волн равна частоте быстроходного вала n_6); K_{d1} – коэффициент делительного диаметра гибкого колеса (при $d_1 \leq 130$ мм $K_{d1} = 1,25$; при $d_1 > 130$ мм $K_{d1} = 1$).

Для стальных зубчатых колес диапазон допускаемых напряжений составляет $[\sigma]_{\text{см}} = 10 \dots 20$ МПа, для пластмассовых – $[\sigma]_{\text{см}} = 3 \dots 5$ МПа.

2.7.5.4. Проектировочный расчет волновой передачи

Для выполнения расчета применяют упрощенные экспериментально проверенные зависимости. Они верны только для эвольвентных зубчатых колес, нарезанных стандартным инструментом с исходным контуром, имеющим угол

зацепления $\alpha = 20^\circ$, коэффициент высоты зуба $h_a^* = 1$, коэффициент радиально-го зазора $C^* = 0,25$ для модулей более 1 мм и $C^* = 0,35$ для модулей до 1 мм.

Делительный диаметр гибкого колеса, мм

$$d_1 = \sqrt[3]{\frac{10T_T K}{\psi_{bd}[\sigma]_{\text{см}}}}, \quad (2.156)$$

где T_T – в ньютонметрах; K – коэффициент режима работы передачи (при спокойной нагрузке с кратностью перегрузки $T_{\text{max}}/T \leq 1,2$ $K = 1$; при умеренной динамической нагрузке с $T_{\text{max}}/T \leq 1,6$ $K = 1,25$; при резко динамической нагрузке с $1,6 < T_{\text{max}}/T \leq 2,5$ $K = 1,75$); ψ_{bd} – коэффициент ширины зубчатого венца (для силовых ММ $\psi_{bd} = 0,15 \dots 0,20$; для малонагруженных кинематических ММ $\psi_{bd} = 0,06 \dots 0,15$).

Ширина зубчатого венца гибкого колеса равна

$$b_1 = \psi_{bd} d_1, \quad (2.157)$$

где ψ_{bd} – коэффициент ширины зубчатого венца (для силовых передач $\psi_{bd} = 0,15 \dots 0,20$; для малонагруженных кинематических передач $\psi_{bd} = 0,06 \dots 0,15$).

При кулачковом генераторе волн делительный диаметр предварительно принимают равным внутреннему диаметру D гибкого колеса, который считают приблизительно равным наружному диаметру гибкого подшипника (см. табл. 2.27).

Числа зубьев z_1, z_2 – см. формулы (2.153), (2.154).

Модуль зубьев равен

$$m = \frac{d_1}{z_1}. \quad (2.158)$$

После вычисления по (2.158) необходимо принять ближайшее стандартное значение модуля.

Уточняют делительный диаметр гибкого колеса в передаче с *дисковым генератором волн*:

$$d_1 = mz_1. \quad (2.159)$$

Находят фактическое передаточное число передачи в зависимости от того, какое колесо вращается:

- при подвижном гибком колесе

$$u_\phi = \frac{z_1}{K_z v}; \quad (2.160)$$

- при подвижном жестком колесе

$$u_\phi = \frac{z_1}{K_z v} + 1. \quad (2.161)$$

Вычисляют отклонение фактического передаточного числа от требуемого и проверяют соответствие отклонения техническим требованиям к ММ. В случае сверхнормативного отклонения необходимо изменить z_1 и скорректировать значение d_1 .

Число зубьев жёсткого колеса

$$z_2 = z_1 + K_z v. \quad (2.162)$$

Толщина зубчатого венца гибкого колеса

$$h_1 = (70 + 0,5u)mz_1 \cdot 10^{-4}, \text{ мм.} \quad (2.163)$$

Толщина оболочки гибкого колеса

$$h_2 = (0,5 \dots 0,8)h_1. \quad (2.164)$$

Относительный боковой зазор между зубьями ненагруженной передачи

$$\bar{j}_{\max} = \frac{j_{\max}}{m} = \frac{T_{\max} b}{d_1^2 h_2 G m} + 4 \cdot 10^{-4} (u - 60), \quad (2.165)$$

где j_{\max} – необходимый боковой зазор между зубьями ненагруженной передачи, мм; G – модуль упругости второго рода для материала гибкого колеса (для стали $G = 8,1 \cdot 10^4$ МПа).

Относительное радиальное упругое деформирование гибкого колеса

$$w_0^* = \frac{w_0}{m} = 0,89 + 8 \cdot 10^{-4} z_1 + 2j_{\max}, \quad (2.166)$$

где w_0 – радиальное упругое деформирование гибкого колеса, мм.

Коэффициент смещения исходного контура зубьев гибкого колеса

$$x_1 = \frac{1,35 - w_0^*}{\frac{0,85}{\sqrt[3]{z_1}} - 0,04}. \quad (2.167)$$

Коэффициент смещения исходного контура зубьев жесткого колеса

$$x_2 = x_1 + w_0^* - 1. \quad (2.168)$$

Относительная глубина захода зубьев

$$h_d^* = \frac{h_d}{m} = 4w_0^* - (4,6 - 4w_0^*)z_1 \cdot 10^{-3} - 2,48, \quad (2.169)$$

где h_d – глубина захода зубьев, мм.

Диаметр окружности впадин зубьев гибкого колеса

$$d_{f1} = m(z_1 - 2h_d^* - 2c^* + 2x_1). \quad (2.170)$$

Диаметр окружности вершин зубьев гибкого колеса

$$d_{a1} = d_{f1} + 2m(h_d^* + c^*). \quad (2.171)$$

Диаметр окружности вершин зубьев жесткого колеса

$$d_{a2} = d_{a1} - 2m(h_d^* + w_0^*). \quad (2.172)$$

Диаметр окружности впадин зубьев жесткого колеса

$$d_{f2} = d_{a1} + 2m(0,15 + w_0^*). \quad (2.173)$$

Для гибкого колеса передачи с кулачковым генератором волн уточняют значение толщины венца по формуле

$$h_1 = 0,5(d_{f1} - D). \quad (2.174)$$

Длина гибкого колеса с дном (см. рис. 2.18, а) $L = 0,8D$, со шлицами (рис. 2.18, в) $L = 0,7D$.

Параметры остальных частей гибкого колеса: $h_3 = (0,7 \dots 1)h_2$; $h_4 = 2h_1$; $h_5 \geq 0,16D$; $c = 0,2b$.

Ширина зубчатого венца жесткого колеса

$$b_2 = b + 0,6\sqrt{b}. \quad (2.175)$$

Длина шлицев (рис. 2.18, в) $b_3 = 0,5b$.

Определяют основные *геометрические параметры генератора волн*.

Геометрию кулачкового генератора (см. рис. 2.16) определяет радиус-вектор поперечного сечения ρ .

Радиус-вектор кулачка в каждой четверти равен

$$\rho = 0,5d_{\text{п}} + WmK_{\alpha}, \quad (2.176)$$

где $d_{\text{п}}$ – внутренний диаметр гибкого подшипника; W – коэффициент радиальной деформации (табл. 2.29); K_{α} – коэффициент, зависящий от угла зацепления α ($K_{\alpha} = 1$ для $\alpha = 20^\circ$; $K_{\alpha} = 0,89$ для $\alpha = 30^\circ$).

Срок службы стандартных гибких подшипников составляет 10 000 ч, при эксплуатации допускаются кратковременные двукратные перегрузки.

Таблица 2.29

Значения коэффициента W

Угол φ	0	10°	20°	30°	40°	50°	60°	70°	80°	90°
W	-1,25	-1,13	-0,91	-0,56	-0,15	0,26	0,57	0,76	0,87	0,9

Эксцентриситет дискового генератора волн (см. рис. 2.17) равен

$$e = 3,4w_0^*m. \quad (2.177)$$

Диаметр дисков

$$D_d = D + 2w_1 - 2e, \quad (2.178)$$

где w_1 – максимальная упругая деформация гибкого колеса с учетом податливости генератора волн и жесткого колеса, а также отклонений размеров от номинальных при изготовлении.

Значение w_1 определяют по формуле

$$w_1 = w_0^* m (0,97 + 0,025 \sqrt{\sigma_{\text{см}}}), \text{ мм.} \quad (2.179)$$

2.7.5.5. Проверочные расчеты

Запас усталостной прочности гибкого колеса

$$s = \frac{0,286 \sigma_{-1} u_{\phi} d_1}{K_d K_{\sigma} w_0^* E h_1 \sqrt{1 + 0,15 \left(\frac{d_1}{L} + \frac{T_r u_{\phi}}{w_0^* d_1 E h_1^2} \right)}}, \quad (2.180)$$

где $E = 2,1 \cdot 10^5$ МПа – модуль упругости первого рода для стали; σ_{-1} – предел выносливости материала гибкого колеса (см. табл. 2.28); K_d – коэффициент увеличения напряжения от сил в зацеплении (формула (2.191)); K_{σ} – эффективный коэффициент концентрации напряжений у основания зуба (формула (2.192)).

Условие обеспечения необходимой выносливости гибкого колеса

$$s \geq [s], \quad (2.181)$$

где $[s] = 1,2$ – минимальный допускаемый запас усталостной прочности.

Величину коэффициента K_d определяют по формуле

$$K_d = 1 + \frac{2,2T_r}{10^8 d_1 h_1^2}, \quad (2.182)$$

а коэффициента K_σ – по формуле

$$K_\sigma = 1 + \frac{0,05 \sqrt{\frac{m d_1 (h_a^* + c^*)}{R_{\min} h_1}} (R_{\min} + 0,01)}{R_{\min} + 0,02}, \quad (2.183)$$

где R_{\min} – минимальный радиус переходной поверхности, равный

$$R_{\min} = \frac{m(h_a^* + c^* - x_1 - \rho^*)}{h_a^* + c^* - x_1 - \rho^* + 0,5z_1} + m\rho^*. \quad (2.184)$$

Значения входящих в (2.183) величин: $c^* = 0,25$ и $\rho^* = 0,4$ при $m > 1$ мм; $c^* = 0,35$ и $\rho^* = 0,4$ при $m = 1 \dots 0,5$ мм; $c^* = 0,5$ и $\rho^* = 0,33$ при $m < 0,5$ мм.

Вероятность P неразрушения гибкого зубчатого колеса в зависимости от коэффициента запаса усталостной прочности приведена в табл. 3.20.

Таблица 2.30

Вероятность P неразрушения гибкого зубчатого колеса и коэффициент K_L вероятности неразрушения гибкого подшипника

s	1,80	1,70	1,60	1,55	1,50	1,45	1,40	1,30	1,20
$P, \%$	99,8	99,6	99,0	98,5	97,8	96,7	95,1	90,0	87,0
K_L	0,50	0,60	0,66	0,69	0,73	0,80	0,90	1,00	

Уточняют КПД передачи. Потери в волновой передаче складываются из потерь в зацеплении и потерь при вращении генератора волн. Сопротивлением подшипников быстроходного и тихоходного валов пренебрегают. В этом случае КПД равен

$$\eta = \left[1 + \frac{1,11vf_1}{\cos^2 \alpha} + 0,3uf_2 \operatorname{tg}(\alpha + \rho') \right]^{-1}, \quad (2.185)$$

где $f_1 = 0,03 \dots 0,05$ – коэффициент, учитывающий трение в зубчатом зацеплении; $f_2 = 0,0015 \dots 0,003$ – условный коэффициент, учитывающий трение во всех элементах генератора волн; $\rho' = \operatorname{arctg} f_1$ – приведенный угол трения.

После уточнения КПД выполняют *тепловой расчет* передачи. Температуру масла в корпусе редуктора определяют по формуле (2.134), причем коэффициент продолжительности работы редуктора находят по формуле

$$K_t = \frac{t_p}{t_{ц}}, \quad (2.186)$$

где t_p – время работы редуктора в течение цикла; $t_{ц}$ – длительность цикла, а поверхность охлаждения редуктора принимают равной

$$A = (8 \dots 10)d_1^2, \text{ м}^2. \quad (2.187)$$

Полученную температуру сравнивают с допускаемой.

Контрольные вопросы

1. Поясните термины: механическая передача, вариатор, редуктор, мультипликатор.
2. Дайте сравнительную характеристику передач трением и передач зацеплением.
3. Перечислите разновидности ремённых передач.
4. Перечислите разновидности и дайте сравнительную характеристику цилиндрических зубчатых передач.
5. Укажите основные геометрические характеристики цилиндрической прямозубой передачи.

6. Какие материалы используются для изготовления цилиндрических зубчатых передач?

7. Поясните формулу: $a'_{вт} = 430(u_t \pm 1) \sqrt[3]{\frac{T_{2т} K_{H\beta т}}{u_t^2 \Psi_{баг} [\sigma_H]_т^2}}$.

8. Перечислите разновидности и дайте сравнительную характеристику конических зубчатых передач.

9. Укажите основные геометрические характеристики конической прямозубой передачи.

10. Перечислите разновидности и дайте сравнительную характеристику червячных передач.

11. Укажите основные геометрические характеристики червячной передачи архимедовым червяком.

12. С какой целью выполняется тепловой расчёт червячной передачи?

13. Изобразите кинематическую схему планетарной передачи $2K-H$.

14. Назовите достоинства и недостатки волновой зубчатой передачи.

3. ПОДШИПНИКИ

3.1. Классификация подшипников по виду трения

Подшипники служат опорами валов, вращающихся осей и других деталей, совершающих вращательное движение. Они воспринимают нагрузки, приложенные к вращающимся деталям, и сохраняют заданное положение осей вращения. От качества подшипников во многом зависит работоспособность и долговечность ММ. При выборе подшипников следует принимать во внимание, что они относятся к наименее долговечным узлам механизмов. Часто именно необходимость технического обслуживания или замены подшипников является причиной остановки ММ.

По виду трения различают *подшипники скольжения* и *подшипники качения*. Кроме того, подшипники скольжения разделяются на опоры *жидкостного*, *полужидкостного* и *полусухого трения*. При жидкостном трении рабочие поверхности деталей разделены слоем смазки, толщина которого больше суммы высот шероховатости поверхностей. Это условие не соблюдается в опорах полужидкостного и полусухого трения, поэтому вращение деталей в них сопровождается износом поверхностей даже без попадания абразивных частиц извне.

Практически во всех ответственных узлах ММ используются подшипники качения, что обусловлено следующими их *преимуществами* перед подшипниками скольжения:

- условный коэффициент трения подшипников качения мал, он равен 0,0015...0,0060 и приближается к коэффициенту жидкостного трения (0,001...0,005);
- подшипники качения не требуют большого количества смазки, что позволяет значительно упростить смазочную систему;
- конструкции подшипников качения позволяют производить их в массовых масштабах и обеспечивать их взаимозаменяемость;
- относительно меньшие осевые габариты.

К недостаткам подшипников качения следует отнести:

- отсутствие конструкций, разъемных в радиальном направлении;
- сравнительно большие радиальные габариты;
- ограниченную быстроходность, связанную с кинематикой и динамикой тел качения;
- низкую работоспособность в агрессивных средах, а также при вибрационных и ударных нагрузках.

Несмотря на сокращение области применения подшипников скольжения, в ряде случаев они остаются незаменимыми, и конструктор обязан это учитывать в процессе разработки ММ.

Далее будут рассмотрены подшипники качения.

3.2. Конструкции и классификация подшипников качения

В общем случае подшипник качения состоит из *колец* с беговыми дорожками, *тел качения* и *сепаратора*. Тела качения расположены между кольцами и перемещаются по беговым дорожкам. Сепаратор представляет собой элемент, охватывающий тела качения и распределяющий их равномерно по окружности.

Кроме того, конструкция подшипника может включать в себя другие детали: уплотнительные и защитные шайбы, конические втулки с круглыми гайками и т. п. В некоторых преобразователях движения используют, наоборот, упрощенные подшипники (без одного из колец). Со всем многообразием подшипников качения можно ознакомиться с помощью специальной технической литературы.

По виду тел качения различают *шариковые* и *роликовые* подшипники. Роликовые делятся на подшипники с *короткими цилиндрическими роликами*, с *длинными цилиндрическими роликами*, *игольчатые*, с *коническими роликами* и с *бочкообразными роликами*.

По направлению воспринимаемой нагрузки подшипники подразделяются на *радиальные*, *упорные*, *радиально-упорные* и *упорно-радиальные*.

Радиальные шариковые подшипники – наиболее простые и дешевые. Они допускают небольшие перекосы вала (до $0,25^\circ$) и могут воспринимать не только радиальные, но и значительные осевые нагрузки. Благодаря своим достоинствам эти подшипники широко распространены в преобразователях движения ММ.

Радиальные роликовые подшипники с короткими цилиндрическими роликами способны воспринимать значительно бóльшие радиальные нагрузки, чем шариковые, но не допускают радиальных нагрузок и перекосов вала.

Радиально-упорные шариковые подшипники имеют особую форму наружного кольца, вследствие чего равнодействующая сил давления шарика на кольцо образует с диаметральной плоскостью угол α , называемый *углом контакта*.

В радиально-упорных и упорно-радиальных роликовых подшипниках используются конические ролики. Сравнение этих подшипников с радиально-упорными шариковыми показывает, что роликовые обладают существенно большей нагрузочной способностью, но гораздо хуже воспринимают перекосы вала.

Шариковые и роликовые упорные подшипники при малых размерах обладают весьма высокой нагрузочной способностью, но воспринимают только осевые нагрузки и не допускают перекоса вала.

Особую группу образуют шариковые и роликовые *радиальные сферические подшипники*. Наружное кольцо такого подшипника имеет сферическую рабочую поверхность, с которой контактируют шарики или бочкообразные ролики. Сферические подшипники способны работать при перекосах вала до $2...3^\circ$ и обеспечивать его вращение в случае установки опор в отдельных корпусах.

По габаритам подшипники разделяют на семь серий диаметров и ширин: *сверхлегкую, особо легкую, легкую широкую, среднюю, среднюю широкую, тяжелую*. Однотипные подшипники с одинаковыми диаметрами внутреннего кольца, но относящиеся к разным сериям, имеют различные размеры тел качения, и, как следствие, различную нагрузочную способность.

По классам точности различают подшипники: класса 0 (*нормальной точности*); 6 (*повышенной*); 5 (*высокой*); 4 (*особо высокой*); 2 (*сверхвысокой*). От точности существенно зависит работоспособность подшипника, но одновременно с повышением точности возрастает и стоимость. Так, с переходом от класса 0 к классу 2 относительная стоимость подшипника повышается примерно в 10 раз. Поэтому использовать высокоточные подшипники в преобразователях движения ММ рекомендуется *в обоснованных случаях*.

3.3. Методика расчета подшипников качения

3.3.1. Исходные данные для расчета

В результате расчета передач и конструирования валов (см. раздел 4) редуктора определены следующие данные, используемые как исходные при расчете подшипников:

- расчетные схемы валов с внешними нагрузками и реакциями опор;
- диаметры опорных поверхностей валов;
- частоты вращения валов;
- срок службы ММ;
- нагрузочная диаграмма (циклограмма);
- особые требования (например, максимальные допустимые значения кинематической погрешности и мертвого хода).

Кроме указанных, из технического задания на редуктор могут следовать дополнительные требования, например, максимальные допустимые габаритные размеры ММ, наличие и кратность динамических нагрузок и т. д.

3.3.2. Предварительный выбор подшипников

Подшипники предварительно назначают по *виду передачи, функции вала* в редукторе и *диаметру опорных поверхностей* $d_{п}$.

Валы цилиндрических зубчатых передач, как правило, устанавливают на шариковых радиальных подшипниках. Для конических и червячных передач необходимы радиально-упорные подшипники; в большинстве случаев в преоб-

разователях движения ММ применяют роликовые радиально-упорные конические подшипники. В кулачковых генераторах волновых передач используют гибкие шариковые подшипники. В тех случаях, когда на вал действуют значительные осевые силы, в подшипниковые узлы включают упорные подшипники.

Для установки быстроходного и промежуточного валов трехосного цилиндрического редуктора могут быть рекомендованы подшипники средней серии, тихоходного вала – легкой. В соосном цилиндрическом редукторе, как правило, опоры всех валов требуют подшипников средней или тяжелой серий. Обычно в обеих опорах вала располагают одинаковые подшипники.

Для валов червяков и шестерен конических передач целесообразно предварительно назначить роликовые радиально-упорные конические подшипники средней серии. Следует иметь в виду, что в передачах с постоянным направлением осевой силы менее нагруженный подшипник часто принимают более легкой серии. Опоры валов червячных колес в большинстве случаев ставят на подшипники легкой серии.

По приведенным выше рекомендациям и значению диаметра $d_{\text{п}}$ выбирают подшипники и выписывают из соответствующего стандарта их основные характеристики.

Предварительно принятые подшипники требуют проверки. Поскольку в подавляющем большинстве преобразователей движения ММ валы вращаются с частотами более 1 об/мин, то далее рассматривается проверка *по динамической грузоподъемности*.

3.3.3. Проверочный расчет подшипников по динамической грузоподъемности

3.3.3.1. Шариковые радиальные подшипники

Вал установлен в двух опорах – A и B , радиальные реакции опор – соответственно R_A и R_B , осевая реакция опоры A – R_{Aa} .

Определяют отношения

$$\frac{R_{Aa}}{VR_A}; \quad (3.1)$$

$$\frac{R_{Aa}}{C_{0r}}, \quad (3.2)$$

где V – коэффициент вращения ($V = 1$ – при вращающемся внутреннем кольце; $V = 1,2$ – при вращающемся наружном кольце подшипника); C_{0r} – табличная статическая грузоподъемность подшипника, Н.

По значению (3.2) и табл. 3.1 устанавливают коэффициент влияния осевого нагружения e .

Таблица 3.1

Коэффициенты e и Y для шариковых радиальных однорядных подшипников

R_{Aa}/C_{0r}	0,014	0,028	0,056	0,084	0,110	0,170	0,280	0,420	0,560
e	0,19	0,22	0,26	0,28	0,30	0,34	0,38	0,42	0,44
Y	2,30	1,99	1,71	1,55	1,45	1,31	1,15	1,04	1,00

Сравнение величин (3.1) и e показывает *значимость осевой силы*. Если $\frac{R_{Aa}}{VR_A} \leq e$, то в дальнейшем осевую составляющую реакции опоры A не учитывают и считают подшипник нагруженным только радиально.

Следующим шагом определяют *эквивалентные нагрузки* на подшипники.

Простейшим случаем нагружения преобразователя движения является *постоянный режим*, при котором величины реакций опор не изменяются со временем. Пусть в опоре A учет осевой реакции необходим. Тогда эквивалентные нагрузки на подшипники равны

$$\begin{aligned} P_{\text{э}A} &= (VXR_A + YR_{Aa})K_{\sigma}K_T; \\ P_{\text{э}B} &= VR_B K_{\sigma} K_T, \end{aligned} \quad (3.3)$$

где $X = 0,56$ – коэффициент радиальной нагрузки; Y – коэффициент осевой нагрузки (см. табл. 3.1); K_6 – коэффициент безопасности (табл. 3.2); K_T – температурный коэффициент (при рабочей температуре подшипника до $125\text{ }^\circ\text{C}$ $K_T = 1$).

Если же осевую реакцию учитывать не нужно, то эквивалентную нагрузку на подшипник A определяют из выражения

$$P_{\text{э}A} = VR_A K_6 K_T. \quad (3.4)$$

Преобразователи движения ММ и роботов часто предназначены для выполнения вполне определенных операций, характеризующихся переменными нагрузками, скоростями движения рабочего органа и продолжительностью. В таких случаях для минимизации массы и размеров ММ необходимо учитывать *переменный характер режима нагружения*.

Учет режима нагружения производится при вычислении *приведенной эквивалентной нагрузки* на подшипник за цикл $P_{\text{эЦ}}$.

Пусть в течение цикла длительностью $t_{\text{Ц}}$ эквивалентная нагрузка на подшипник изменяется линейно от $P_{\text{эmin}}$ до $P_{\text{эmax}}$, а частота вращения вала $n = \text{const}$. Тогда приведенная эквивалентная нагрузка равна

$$P_{\text{эЦ}} = \frac{P_{\text{эmin}} + 2P_{\text{эmax}}}{3}. \quad (3.5)$$

В более сложном случае, когда цикл состоит из k участков с различной нагрузкой на подшипник, приведенная эквивалентная нагрузка равна

$$P_{\text{эЦ}} = \sqrt[3]{\frac{\sum_{i=1}^k (P_{\text{э}i}^3 t_i n)}{t_{\text{Ц}}}}, \quad (3.6)$$

где $P_{\partial i}$ – эквивалентная нагрузка на i -ом участке; t_i – длительность i -ого участка.

В самом общем случае, когда в зависимости от участка циклограммы изменяется не только нагрузка, но и частота вращения вала, приведенная эквивалентная нагрузка равна

$$P_{\partial \text{Ц}} = \sqrt[3]{\frac{\sum_{i=1}^k (P_{\partial i}^3 t_i n_i)}{\sum_{i=1}^k (t_i n_i)}}, \quad (3.7)$$

где n_i – частота вращения вала на i -ом участке циклограммы.

Затем находят *требуемую динамическую грузоподъемность* подшипников по формулам:

для постоянного режима нагружения

$$\begin{aligned} C_{\text{гр}A} &= P_{\partial A} \sqrt[3]{L \cdot 10^{-6}}; \\ C_{\text{гр}B} &= P_{\partial B} \sqrt[3]{L \cdot 10^{-6}}; \end{aligned} \quad (3.8)$$

для переменного режима нагружения

$$\begin{aligned} C_{\text{гр}A} &= P_{\partial \text{Ц}A} \sqrt[3]{L \cdot 10^{-6}}; \\ C_{\text{гр}B} &= P_{\partial \text{Ц}B} \sqrt[3]{L \cdot 10^{-6}}, \end{aligned} \quad (3.9)$$

где L – число оборотов вала за срок службы ММ.

Вид выражения для вычисления L зависит от того, как в техническом задании указан срок службы ММ. Если оговорен календарный срок $T_{\text{сл}}$, лет, то число оборотов вала за $T_{\text{сл}}$ вычисляют следующим образом: для первого и второго из рассмотренных случаев

$$L = 60T_{\text{сл}} n_{\text{рд}} n_{\text{см}} t_{\text{см}} n; \quad (3.10)$$

для третьего случая

$$L = 60T_{\text{сл}} n_{\text{рд}} n_{\text{см}} t_{\text{см}} \sum_{i=1}^k \frac{t_i n_i}{t_{\text{ц}}}, \quad (3.11)$$

где $n_{\text{рд}}$ – число рабочих дней в году; $n_{\text{см}}$ – число рабочих смен в сутки; $t_{\text{см}}$ – число рабочих часов в смену.

В (3.10) и (3.11) $t_{\text{ц}}$ и t_i – в минутах.

Если же задан фонд рабочего времени $T_{\text{раб}}$, то (3.10) и (3.11) соответственно преобразуются в формулы

$$L = 60T_{\text{раб}} n; \quad (3.12)$$

$$L = 60T_{\text{раб}} \sum_{i=1}^k \frac{t_i n_i}{t_{\text{ц}}}. \quad (3.13)$$

Затем сравнивают бóльшую из требуемых динамических грузоподъемностей подшипников с табличной динамической грузоподъемностью $C_{\text{табл}}$ предварительно принятого подшипника. Подшипник проработает заданный срок, если $C_{\text{тр}} \leq C_{\text{табл}}$.

Особо следует рассмотреть проверку подшипников генераторов волн волновой передачи.

В процессе расчета передачи (см. п. 2.7.5) в соответствии с величиной делительного диаметра d_1 гибкого колеса был принят гибкий подшипник кулачкового генератора или подшипники шариковые радиальные дискового генератора. Требуемая динамическая грузоподъемность подшипника равна

$$C_{\text{тр}} = 0,01P_{\text{эЦ}} \frac{\sqrt[3]{60T_{\text{раб}}n_{\text{п}}}}{K_L}, \quad (3.14)$$

где $P_{\text{эЦ}}$ – определяют по формуле (3.7) с подстановкой в нее $P_{\text{э}i}$, найденных по (3.15); $n_{\text{п}}$ – эквивалентная частота вращения подшипника генератора волн (формула (3.16)); K_L – коэффициент, принимаемый по табл. 2.30.

Величину эквивалентной нагрузки на i -ом участке находят по выражению

$$P_{\text{э}i} = 0,6 \frac{T_{\text{т}i}}{d_1} V K_6 K_T, \quad (3.15)$$

где $T_{\text{т}i}$ – крутящий момент на тихоходном валу на i -ом участке; $V = 1,2$; K_6 – принимают в зависимости от вида генератора волн ($K_6 = 1,1$ – для кулачкового генератора; $K_6 = 1,3$ – для дискового генератора); K_T – принимают в зависимости от температуры подшипника ($K_T = 1$ – до 100 °С включительно; $K_T = 1 \dots 1,05$ – от 100 °С до 125 °С; $K_T = 1,05 \dots 1,4$ – от 125 °С до 250 °С).

Эквивалентная частота вращения подшипника равна

$$n_{\text{п}} = \sum_{i=1}^k \frac{t_i n_{\text{т}i}}{t_{\text{Ц}}}, \quad (3.1)$$

а $n_{\text{т}i}$ определяют следующим образом: для кулачкового генератора (рис. 2.16) $n_{\text{т}i} = n_{1i}$; для дискового генератора (рис. 2.17) с подвижным гибким зубчатым колесом

$$n_{\text{т}i} = \frac{d_1(n_{3i} - n_{1i})}{D_{\text{д}}}; \quad (3.17)$$

для дискового генератора с подвижным жестким зубчатый колесом

$$n_{\text{ш}} = \frac{d_1(n_{3i} - n_{2i})}{D_d}; \quad (3.18)$$

Полученное значение требуемой динамической грузоподъемности не должно превышать расчетную грузоподъемность, которую для гибкого подшипника находят по формулам:

при диаметре шарика $d_{\text{ш}} \leq 25,4$ мм

$$C = f_c (j \cos \gamma)^{0,7} z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,8}; \quad (3.19)$$

при диаметре шарика $d_{\text{ш}} > 25,4$ мм

$$C = 3,647 f_c (j \cos \gamma)^{0,7} z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,4}, \quad (3.20)$$

где $f_c = 5 \dots 5,2$ – коэффициент динамической грузоподъемности; j – число рядов шариков в подшипнике (обычно $j = 1$); γ – угол контакта тел качения с кольцами; $z_{\text{ш}}$ – количество шариков в ряду.

Стандартные $z_{\text{ш}}$ и $d_{\text{ш}}$ см. в табл. 2.27.

Осевая нагрузка на гибкий подшипник равна 0,1 радиальной, поэтому

$$\gamma = \arctg 0,1 = 5,71^\circ = 5^\circ 43', \quad (3.21)$$

и формулы (3.19) и (3.20) для $j = 1$ принимают вид

$$C = 0,996 f_c z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,8} \approx f_c z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,8}; \quad (3.22)$$

$$C = 3,63 f_c z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,4}. \quad (3.23)$$

Расчетные динамические грузоподъемности шариковых радиальных подшипников дисковых генераторов волн указаны в стандарте на подшипники.

При выполнении условия $C_{\text{тр}} \leq C$ принятый подшипник будет обладать заданной долговечностью $T_{\text{раб}}$.

Может быть также использован другой способ оценки долговечности гибкого подшипника.

Определяют эквивалентный крутящий момент на тихоходном валу за рабочий цикл

$$T_{\text{э}} = \sum_{i=1}^k \frac{T_{\text{т}i} t_i n_{\text{ш}i}}{t_{\text{ц}}}. \quad (3.24)$$

после чего находят расчетную долговечность подшипника по формуле

$$L_h = \frac{n_{\text{max}} T_{\text{max}}}{n_{\text{п}} T_{\text{э}}} \cdot 10^4, \text{ ч}, \quad (3.25)$$

где n_{max} – предельная частота вращения для подшипника по табл. 2.27; T_{max} – допустимый крутящий момент на тихоходном валу (табл. 3.2).

3.3.3.2. Радиально-упорные подшипники

В радиально-упорных подшипниках контактные линии наклонены к оси вала на угол α , что при радиальном нагружении приводит к появлению *внутренних осевых сил* S (рис. 3.1).

Значения S определяют так:

- для шариковых подшипников

$$S = eR; \quad (3.26)$$

- для роликовых подшипников

$$S = 0,83eR, \quad (3.27)$$

где e – параметр осевой нагрузки (по каталогу).

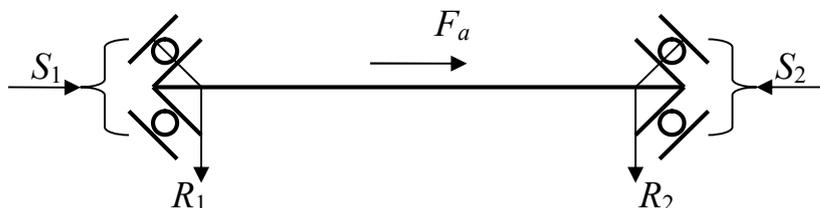


Рис. 3.1. Схема вала на радиально-упорных подшипниках (F_a – внешняя осевая сила)

Осевые нагрузки на подшипники вычисляют по формулам, приведенным в табл. 3.3.

Эквивалентную динамическую нагрузку на роликовый подшипник следует определять по формулам:

$$\begin{aligned} C_{r\text{тр}} &= P_{\text{э}} \sqrt[3,33]{L \cdot 10^{-6}}; \\ C_{r\text{ц}} &= P_{\text{эЦ}} \sqrt[3,33]{L \cdot 10^{-6}}. \end{aligned} \quad (3.28)$$

В остальном методика проверки аналогична приведенной в подпункте 3.3.3.1.

Таблица 3.2

Допустимый крутящий момент на тихоходном валу волновой передачи T_{\max} , Нм

Диаметр наружного кольца D , мм	Передаточное число u	Допустимый крутящий момент T_{\max}
100	80	180
	100	200
	125	224
	160	250
	200 и более	280
120	80	355
	100	400
	125	450
	160	500
	200 и более	560
160	80	710
	100	800
	125	900
	160 и более	1000
200	80	1400
	100	1600
	125	1800
	160 и более	2000
240	80	2800
	100	3150
	125 и более	3550
320	80	5600
	100 и более	6300

Таблица 3.3

Расчет осевых нагрузок на подшипники вала по рис. 3.1

Условия нагружения	Осевые нагрузки
$S_1 > S_2; F_a > 0$	$F_{a1} = S_1;$
$S_1 < S_2; F_a > S_2 - S_1$	$F_{a2} = S_1 + F_a$
$S_1 < S_2; F_a < S_2 - S_1$	$F_{a1} = S_2 - F_a; F_{a2} = S_2$

Контрольные вопросы

1. Дайте сравнительную характеристику подшипников скольжения и подшипников качения.

2. Перечислите разновидности подшипников по виду тел качения.
3. Перечислите разновидности подшипников по направлению воспринимаемой нагрузки.
4. Что является причиной возникновения осевой силы от радиальной нагрузки в радиально-упорном подшипнике?
5. Какую функцию в подшипнике качения выполняет сепаратор?
6. Что такое статическая грузоподъёмность и динамическая грузоподъёмность подшипника?
7. Опишите порядок проверки на долговечность шарикового радиального подшипника.

4. ВАЛЫ И ОСИ

4.1. Общие сведения

Валы и оси служат для размещения на них вращающихся деталей: зубчатых колес, шкивов, барабанов и т.п. Отличие вала от оси состоит в том, что он передает крутящий момент от одной детали к другой, а ось не передает. *Вал* всегда *вращается*, а *ось* может быть как *вращающейся*, так и *не вращающейся*.

Различают валы *прямые*, *коленчатые* и *гибкие*. В преобразователях движения ММ, как правило, применяются прямые валы. Коленчатые и гибкие валы относятся к специальным деталям и далее рассматриваться не будут.

По наличию ступеней валы делятся на *гладкие* и *ступенчатые*. Наличие ступеней связано с установкой на валу деталей и самого вала в опорах. В некоторых случаях выполнение вала или оси ступенчатой формы позволяет существенно уменьшить их массу.

Валы изготавливают *сплошными* или *полыми*. Полость либо уменьшает массу вала, либо позволяет пропустить через вал другую деталь, подвести масло к контактирующим поверхностям и т.п.

Для изготовления валов и осей применяют преимущественно *углеродистые* и *легированные стали*, предусматривающие все возможные виды упрочнения.

Далее приводится методика расчета вала преобразователя движения ММ.

4.2. Методика расчета валов

4.2.1. Исходные данные

Размеры устанавливаемых на вал элементов (зубчатых и червячных колес, посадочные диаметры подшипников и т.п.).

Значения нагрузок на эти элементы. Передаваемый валом крутящий момент.

Частота вращения вала.

Циклограмма нагружения и срок службы ММ.

Дополнительные требования (ориентировочные габаритные размеры, тип подшипников и пр.).

4.2.2. Проектировочный расчет

Определяют *характерный диаметр* вала

$$d = 10^3 \sqrt{\frac{T}{0,2[\tau]}}, \quad (4.1)$$

где T – передаваемый валом крутящий момент, Нм; $[\tau] = 20$ МПа – уменьшенное допускаемое касательное напряжение.

Для быстроходных и тихоходных валов d – диаметр выходного конца, его значение следует принять ближайшее по ГОСТ 12080 или ГОСТ 12081 к полученному по (4.1).

Для промежуточных валов d – диаметр ступеньки под зубчатым (или червячным) колесом, его значение следует принять ближайшее по ряду $Ra40$ к полученному по (4.1).

Диаметры остальных ступенек вала назначают с учетом величины характерного диаметра конструктивно, принимая во внимание требования технологии изготовления и сборки, ряд номинальных размеров, вероятные размеры подшипников и известные размеры сопряженных с валом деталей.

4.2.3. Разработка расчетной схемы

Расчетная схема представляет собой схематичное изображение вала в виде двухопорной балки. Нагрузки показывают в виде сосредоточенных сил и моментов. При необходимости балку снабжают пометками с указанием диаметров ступенек.

Быстроходный вал преобразователя движения ММ соединяют с валом двигателя без компенсирующей муфты, поэтому консольная нагрузка на его выходной конец отсутствует. Часто шестерню быстроходной ступени редуктора

насаживают прямо на вал двигателя, в результате чего необходимость в быстроходном вале отпадает.

На выходной конец тихоходного вала преобразователя движения ММ общего назначения может действовать *консольная нагрузка*, которую учитывают в виде силы, определяемой по следующим формулам:

для одноступенчатого цилиндрического редуктора

$$F_k = 125\sqrt{T_T}, \text{ Н}; \quad (4.2)$$

для двухступенчатого цилиндрического редуктора и для червячного редуктора

$$F_k = 250\sqrt{T_T}, \text{ Н}, \quad (4.3)$$

где T_T – в ньютонметрах.

Длины ступенек при разработке расчетной схемы назначают с учетом размеров деталей, ряда номинальных размеров, а также соотношений, принятых в практике конструирования.

4.2.4. Построение эпюр изгибающих и крутящих моментов в поперечных сечениях вала

Данный пункт расчета подробно рассматривается в курсе «Сопротивление материалов».

4.2.5. Проверка вала на усталостную прочность

Методика проверки достаточно полно изложена в учебной литературе. Здесь же остановимся на нижеследующем отличии проверки вала преобразователя движения ММ от проверки вала редуктора общего назначения.

Разрабатывая преобразователь движения ММ, техническое задание на который содержит требование *обеспечения минимальной массы*, конструктор обязан решить вопрос о возможности и целесообразности корректировки диаметров ступенек вала по результатам проверки.

Пусть в опасном сечении вала диаметром d_{oc} запас усталостной прочности больше допускаемого, т.е. $s > [s]$. Тогда скорректированное значение диаметра достаточно точно может быть определено по формуле

$$d'_{oc} \approx d_{oc} \sqrt[3]{\frac{[s]}{s}}. \quad (4.4)$$

Допускаемый запас прочности, рекомендуемый технической литературой, $[s] = 1,6 \dots 2,1$, но при повышенных требованиях к жесткости следует принимать $[s] = 2,5 \dots 3$. В случае, когда к ММ предъявляются высокие требования в части точности, желательно обеспечить $[s] = 2,5$. В силу различных причин (минимизация массогабаритных показателей, определенная компоновка преобразователя движения и т. д.) $[s]$ может быть понижен до $[s] = 1,6$, но тогда появляется опасность возникновения слишком большой погрешности в передаче из-за деформации вала.

Изменение диаметра одного участка при сохранении принятого соотношения диаметров повлечет за собой соответствующее уменьшение диаметров остальных участков, в том числе – под подшипниками. На предыдущем этапе (подразд. 3.3) принятые предварительно подшипники были проверены по динамической грузоподъемности, их серия, а, возможно, и тип были подобраны таким образом, чтобы максимально приблизить $C_{гр}$ к C_r . Корректировка же диаметров вала делает эти подшипники непригодными, поэтому следует заново назначить подшипники и выполнить их проверку.

Контрольные вопросы

1. В чём отличие вала от оси?
2. Перечислите разновидности валов.
3. С какой целью на валах механических передач выполняют ступеньки?
4. Поясните формулу: $F_k = 125\sqrt{T_t}$.
5. Перечислите пункты методики расчёта вала.

ЗАКЛЮЧЕНИЕ

В учебном пособии представлены конструкции и методы расчёта соединений, передач, подшипников и валов, наиболее часто применяющихся в механической составляющей мехатронных модулей. Правильностью проектирования перечисленных компонентов определяется компактность и надёжность устройства в целом, поэтому задачу, на решение которой направлено учебное пособие, можно считать выполненной.

Необходимо подчеркнуть, однако, что разработка данного учебного материала не имела целью формирование исчерпывающих знаний в области проектирования механического оборудования. Раздел «Валы и оси» дан сокращённо, и для получения более полной информации учащимся следует обратиться к специальным источникам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Анурьев В. И. Справочник конструктора-машиностроителя: В 3-х т. – М.: Машиностроение, 1992.

Дунаев П. Ф., Леликов О. П. Конструирование узлов и деталей машин: учеб. пособие для студ. высш. учеб. заведений. – М.: Изд. центр «Академия», 2008.

Иванов М. П., Финогенов В. А. Детали машин: учебник для академического бакалавриата. – М.: Изд-во Юрайт, 2014.

Таугер В. М. Конструирование преобразователей движения мехатронных модулей: учеб. пособие. – Екатеринбург: Изд-во УрГУПС, 2006.

Таугер В. М. Техническая механика. Детали машин: учеб. пособие – Екатеринбург: Изд-во УГГУ, 2018.

Таугер В. М. Конструирование мехатронных модулей: учеб. пособие. – М: Изд-во «Ай Пи Ар Медиа», 2022.

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА
Б1.В.09.02 ДЕТАЛИ МЕХАТРОННЫХ МОДУЛЕЙ**

Специальность –
21.05.04 Горное дело
Специализация –

Мехатроника и робототехника промышленных комплексов

Авторы: Таугер В.М.

Одобрена на заседании кафедры

Технической механики
(название кафедры)

Зав. кафедрой

Волков Е.Б.
(Фамилия И.О.)

Протокол № 1 от 13.09.2024
(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического
(название факультета)

Председатель

Осипов П.А.
(Фамилия И.О.)

Протокол № 2 от 18.10.2024
(Дата)

Екатеринбург

Задания к курсовому проекту составлены в соответствии с учебным планом по дисциплине «Детали мехатронных модулей» для студентов специальности 21.05.04 Горное дело, специализация «Мехатроника и робототехника промышленных комплексов».

Задания содержат общие требования к курсовому проекту и 6 различных кинематических схем двухступенчатых редукторов. Для каждой схемы предусмотрены 6 вариантов исходных данных.

Составитель: В.М. Таугер, канд. техн. наук, доцент кафедры «Техническая механика».

ОБЩИЕ ТРЕБОВАНИЯ.....	4
2. ЗАДАНИЯ НА КУРСОВОЙ ПРОЕКТ	5
2.1. Коническо-цилиндрический редуктор	5
2.2. Двухступенчатый цилиндрический соосный редуктор	6
2.3. Двухступенчатый цилиндрический трехосный редуктор	6
2.4. Двухступенчатый цилиндрический трехосный редуктор	7
2.5. Двухступенчатый планетарный редуктор	8
2.6. Червячно-цилиндрический редуктор	9
2.7. Графики сменной нагрузки	10
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	11

1. ОБЩИЕ ТРЕБОВАНИЯ

Изучение курса «ДЕТАЛИ МЕХАТРОННЫХ МОДУЛЕЙ» заканчивается курсовым проектом - практической работой, выполняемой студентом под руководством преподавателя.

Темой курсового проекта является проектирование двухступенчатого редуктора. Задание на проект выдается преподавателем каждому студенту индивидуально.

Состав проекта: расчетно-пояснительная записка; сборочный чертеж редуктора; спецификация;

чертежи четырех основных деталей передач.

Расчетно-пояснительная записка оформляется в соответствии со стандартом на текстовые документы [1] на листах формата А4. Она должна содержать обоснование основных конструктивных решений и все расчеты, необходимые для разработки редуктора [2, 3, 4].

Сборочный чертеж редуктора выполняется на одном листе формата А1 в соответствии с требованиями ЕСКД и действующими стандартами.

К сборочному чертежу прилагается спецификация, выполняемая по ЕСКД.

Чертеж каждой детали выполняется на листе формата А3. К основным деталям передач относятся шестерни, зубчатые колеса, червяки, червячные колеса, валы для установки зубчатых или червячных колес. Студент сам выбирает детали, причем необходимо соблюдение следующего условия: две из вычерчиваемых деталей должны составлять зубчатую (а в задании б - червячную) пару, например, шестерня и зубчатое колесо быстроходной ступени. Чертежи должны содержать всю конструкторско-технологическую информацию, необходимую для изготовления деталей в соответствии с их назначением.

Оценка за курсовое проектирование выставляется по результатам проверки проекта преподавателем и защиты его студентом.

2. ЗАДАНИЯ НА КУРСОВОЙ ПРОЕКТ

2.1. Коническо-цилиндрический редуктор

Спроектировать коническо-цилиндрический редуктор, кинематическая схема которого показана на рис. 2.1. Быстроходная ступень – с круговым зубом, тихоходная – косозубая.

Исходные данные – см. табл. 2.1.

При выборе передаточных чисел ступеней использовать рекомендации табл. 2.2. Коэффициент ширины тихоходной ступени принять $\psi_{\text{вн}} = 0,315$.

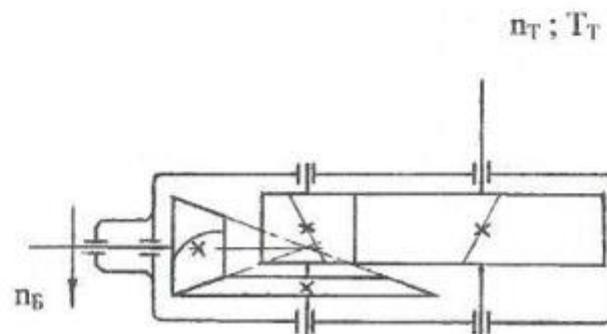


Рис. 2.1

Таблица 2.1

Исходные данные для проектирования коническо-цилиндрического редуктора

Параметр	Вариант					
	1	2	3	4	5	6
Частота вращения быстроходного вала, n_B , об/мин	3000	1500	1000	750	3000	1500
Частота вращения тихоходного вала, n_T , об/мин	130	85	63	54	270	188
Крутящий момент на тихоходном валу, T_T , Нм	1600	800	1250	1000	1400	1800
График сменной нагрузки по рис. 2.7	а	б	в	г	д	а
Срок службы, лет	3	5	8	3	5	8

Таблица 2.2

Передаточные числа быстроходной ступени

Передаточное число редуктора u	8; 9	10; 11,2	12,5; 14	16	≥ 18
Передаточное число быстроходной ступени u_B	3,15	3,55	4	4,5	5

2.2. Двухступенчатый цилиндрический соосный редуктор

Спроектировать двухступенчатый цилиндрический соосный редуктор, кинематическая схема которого показана на рис. 2.2. Обе ступени косозубые.

Исходные данные – см. табл. 2.3.

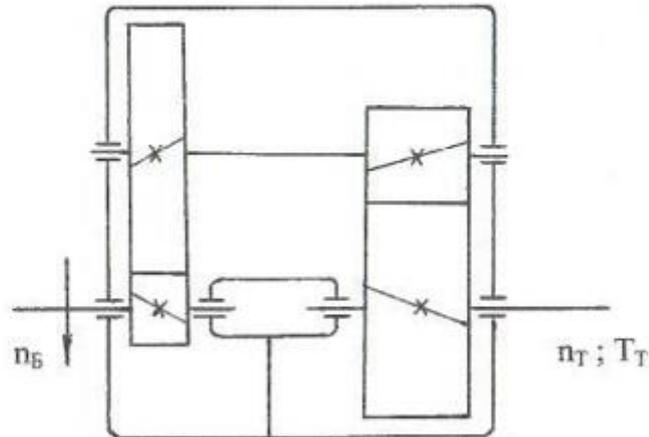


Рис. 2.2

Таблица 2.3

Исходные данные для проектирования двухступенчатого цилиндрического соосного редуктора

Параметр	Вариант					
	1	2	3	4	5	6
Частота вращения быстроходного вала, n_B , об/мин	750	1000	1500	3000	1500	3000
Частота вращения тихоходного вала, n_T , об/мин	67	71	75	67	60	75
Крутящий момент на тихоходном валу, T_T , Нм	1000	1600	1250	800	1800	1400
График сменной нагрузки по рис. 2.7	а	б	в	г	д	а
Срок службы, лет	8	5	4	3	5	3

2.3. Двухступенчатый цилиндрический трехосный редуктор

Спроектировать двухступенчатый цилиндрический трехосный редуктор, кинематическая схема которого показана на рис. 2.3. Обе ступени косозубые.

Исходные данные – см. табл. 2.4.

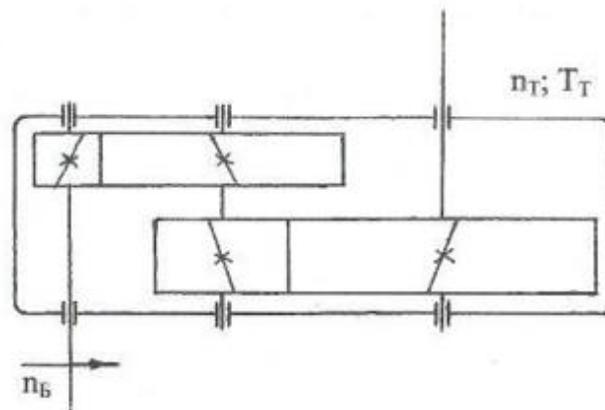


Рис. 2.3

Таблица 2.4

Исходные данные для проектирования двухступенчатого цилиндрического трехосного редуктора

Параметр	Вариант					
	1	2	3	4	5	6
Частота вращения быстроходного вала, n_b , об/мин	750	1000	1500	3000	1500	3000
Частота вращения тихоходного вала, n_r , об/мин	67	71	75	85	60	110
Крутящий момент на тихоходном валу, T_r , Нм	1000	1600	1250	800	1800	1400
График сменной нагрузки по рис. 2.7	а	б	в	г	д	а
Срок службы, лет	8	5	4	3	5	3

2.4. Двухступенчатый цилиндрический трехосный редуктор

Спроектировать двухступенчатый цилиндрический трехосный редуктор, кинематическая схема которого показана на рис. 2.4. Быстроходная ступень – разнесенный шеврон.

Исходные данные – см. табл. 2.4.

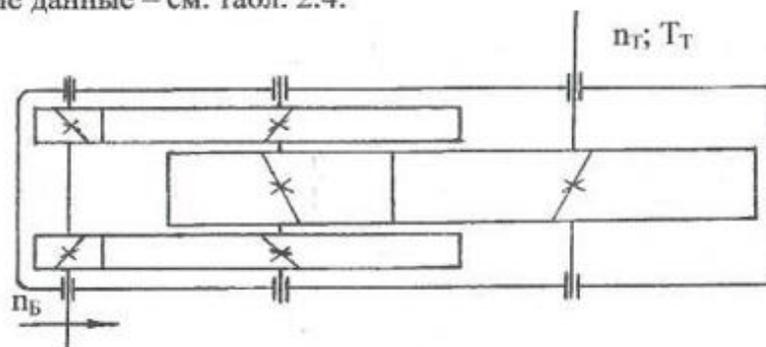


Рис. 2.4

2.5. Двухступенчатый планетарный редуктор

Спроектировать двухступенчатый планетарный редуктор, кинематическая схема которого показана на рис. 2.5. Обе ступени прямозубые.

Исходные данные – см. табл. 2.5.

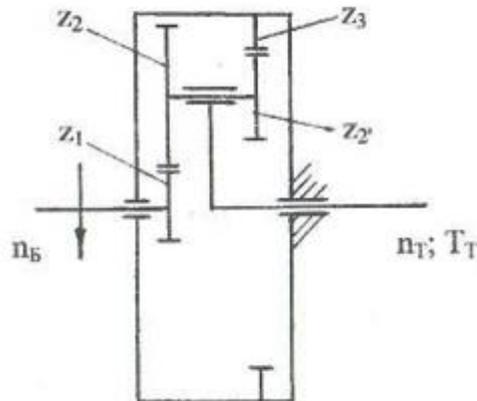


Рис. 2.5

Рекомендуемые числа зубьев указаны в табл. 2.6. Число сателлитов для всех вариантов равно трем.

Таблица 2.5

Исходные данные для проектирования двухступенчатого планетарного редуктора

Параметр	Вариант					
	1	2	3	4	5	6
Частота вращения быстроходного вала, n_B , об/мин	750	1000	1500	3000	1500	3000
Частота вращения тихоходного вала, n_T , об/мин	60	71	75	188	150	170
Крутящий момент на тихоходном валу, T_T , Нм	1000	1600	1250	800	1800	1400
График сменной нагрузки по рис. 2.7	а	б	в	г	д	а
Срок службы, лет	8	5	4	3	5	3

Таблица 2.6

Рекомендуемые числа зубьев колес планетарного редуктора

Стандартное передаточное число редуктора u	10	12,5	14	16	18	20
z_1	18					
z_2	54		64	54	63	72
z_2'	36	27	32	18	21	24
z_3	108	99	114	90	102	114

2.6. Червячно-цилиндрический редуктор

Спроектировать червячно-цилиндрический редуктор, кинематическая схема которого показана на рис. 2.6. Быстроходная ступень – червячная с архимедовым червяком, тихоходная – цилиндрическая косозубая.

Исходные данные – см. табл. 2.7.

Таблица 2.7

Исходные данные для проектирования червячно-цилиндрического редуктора

Параметр	Вариант					
	1	2	3	4	5	6
Частота вращения быстроходного вала, n_B , об/мин	750	1000	1500	3000	1500	3000
Частота вращения тихоходного вала, n_T , об/мин	26	25	30	24	19	48
Крутящий момент на тихоходном валу, T_T , Нм	1000	1600	1250	1400	1800	1120
График сменной нагрузки по рис. 2.7	а	б	в	г	д	а
Срок службы, лет	8	5	4	3	5	3

В редукторах с передаточным числом $u \leq 50$ принимать передаточное число быстроходной ступени $u_B = 8$.

В редукторах с $u > 50$ принимать передаточное число тихоходной ступени $u_T = 6,3$.

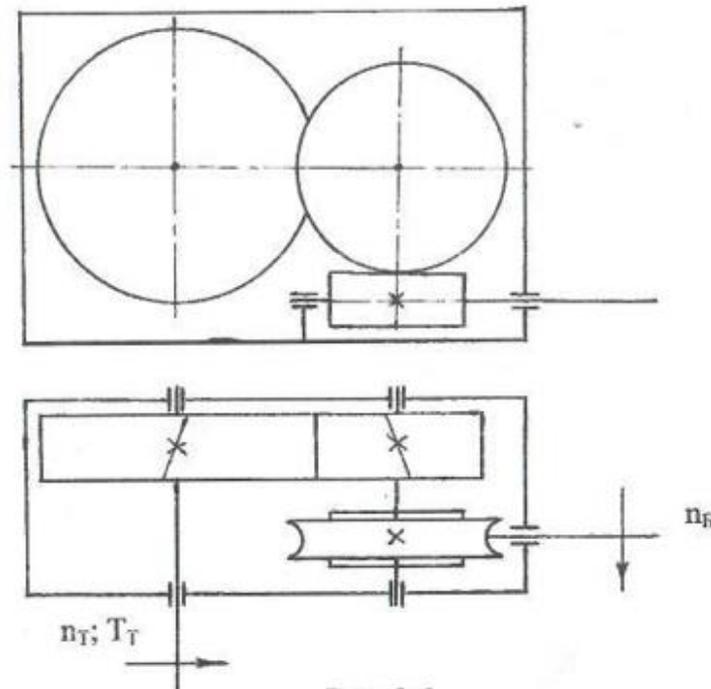


Рис. 2.6

2.7. Графики сменной нагрузки

Графики нагрузки редукторов в течение смены приведены на рис. 2.7.

Обозначения: $T_{\text{ном}}$ – номинальный крутящий момент; $T_{\text{пик}} = 2,5T_{\text{ном}}$ – пиковый момент; T_1, T_2, T_3 – значения нагрузки в течение смены; $t_{\text{см}}, t_1, t_2, t_3$ – продолжительность действия нагрузки соответствующей величины в течение смены; $n_{\text{см}}$ – число рабочих смен в сутки. Число рабочих дней в году $n_{\text{рд}} = 240$.

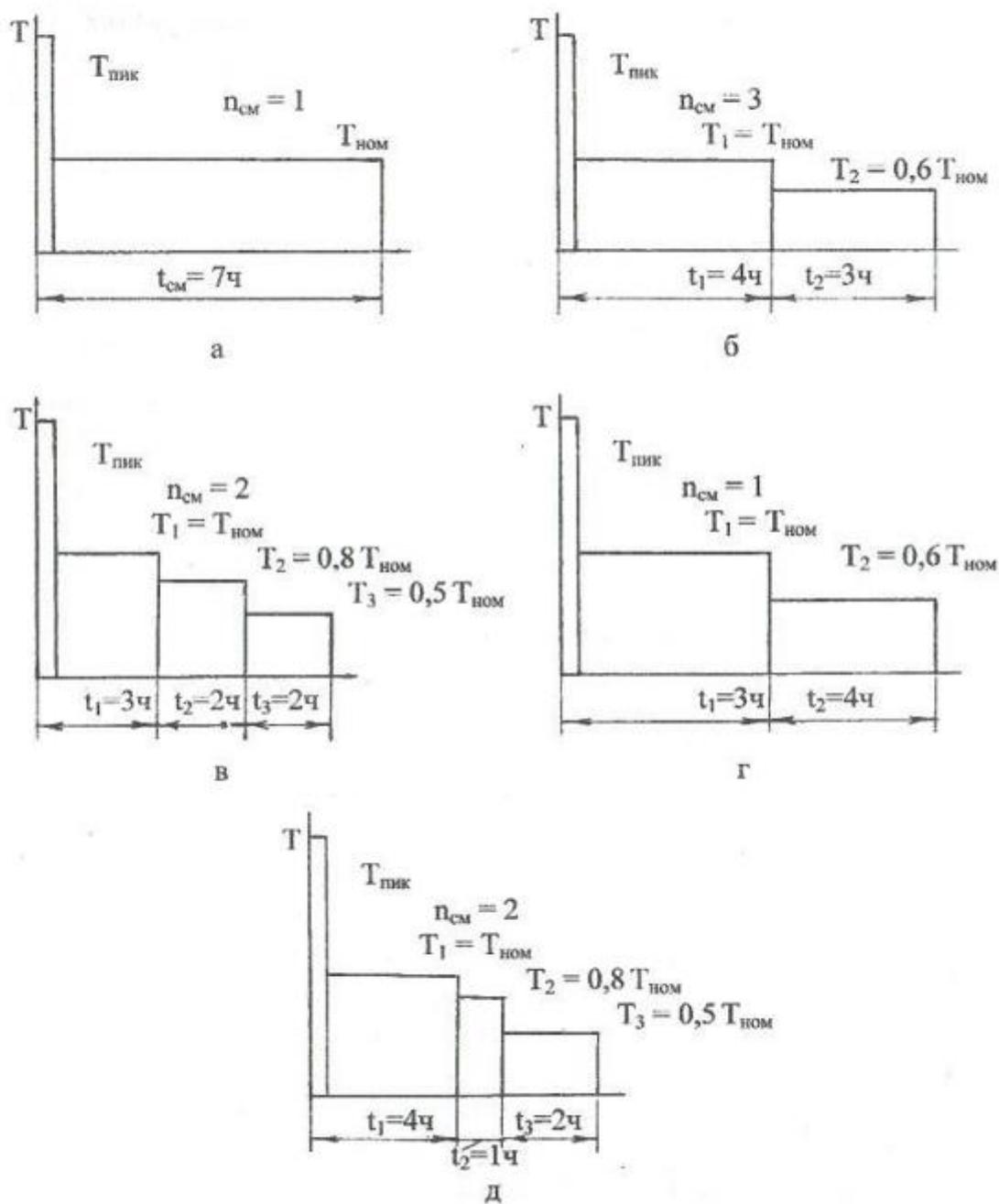


Рис. 2.7

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 2.106-96. ЕСКД. Текстовые документы. – М.: Изд-во стандартов, 1997.
2. Дунаев П.Ф., Леликов О.П. Конструирование узлов и деталей машин. – М.: Высш. шк., 1998.
3. Таугер В.М., Ахлюстина Н.В. Расчет и курсовое проектирование деталей машин. Ч. 2 – Екатеринбург: Изд-во УрГУПС, 2005.
4. Шейнблит А.Е. Курсовое проектирование деталей машин. – М.: Высш. шк., 1991.

МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «Уральский государственный горный университет»



УТВЕРЖДАЮ
И. о. Проректора по учебно-методической
работе _____ В. В. Зубов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА Б1.В.09.02 ДЕТАЛИ МЕХАТРОННЫХ МОДУЛЕЙ

Специальность –
21.05.04 Горное дело
Специализация –

Мехатроника и робототехника промышленных комплексов

Авторы: Таугер В.М.

Одобрена на заседании кафедры

Технической механики
(название кафедры)

Зав. кафедрой

Волков Е.Б.
(Фамилия И.О.)

Протокол № 1 от 13.09.2024
(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического
(название факультета)

Председатель

Осипов П.А.
(Фамилия И.О.)

Протокол № 2 от 18.10.2024
(Дата)

Екатеринбург

Изложены современные методы расчёта и конструирования деталей машин, содержатся сведения, необходимые для курсового проектирования. Подробно рассмотрены выбор электродвигателя, определение передаточного числа и разбивка его по ступеням, вычисление кинематических и силовых параметров привода. Приводятся рекомендации по проектному и проверочному расчёту зубчатых передач, проектному и проверочному расчёту валов, подбору подшипников по динамической грузоподъёмности и проверочному расчёту шпонок.

Методические материалы предназначены для студентов специальности для студентов специальности 21.05.04 Горное дело, специализация «Мехатроника и робототехника промышленных комплексов».

1. РАСЧЁТ ЗАКРЫТЫХ ЦИЛИНДРИЧЕСКИХ ПЕРЕДАЧ

1.1 Общие положения

Зубчатые передачи составляют наиболее распространенную и важную группу механических передач. Их применяют в широком диапазоне областей и условий работы: от приборов до самых тяжелых машин, для передачи крутящих моментов до миллиона Нм, с диаметром колес от долей миллиметра до десяти и более метров. Наибольшее распространение имеют передачи с цилиндрическими колесами, как наиболее простые в изготовлении и эксплуатации, надежные и малогабаритные. Методика расчета на прочность эвольвентных цилиндрических передач, рассматриваемая в данном пособии, основана на ГОСТ 21354-87. При этом введены некоторые упрощения, мало влияющие на результаты расчетов для большинства случаев практики. Так, например, в расчетных формулах не учитываются некоторые коэффициенты, равные единице или очень близкие к ней.

За основную систему в методическом пособии принята Международная система СИ.

Объектами курсового проектирования обычно являются приводы различных машин и механизмов, использующие большинство деталей и узлов общемашиностроительного применения.

Студент, получив индивидуальное задание на курсовой проект, должен, прежде всего, ознакомиться с назначением, устройством, принципом действия и условиями работы той машины, привод которой он должен рассчитать и спроектировать. Затем он должен составить кинематическую схему привода или всего механизма.

После этого студент переходит к расчету привода. При этом рекомендуют следующий порядок расчета:

- 1) выбор электродвигателя по требуемой номинальной мощности, заданной частоте вращения и условиям работы;
- 2) определение общего передаточного числа привода и разбивка его по ступеням, если редуктор двух - или трехступенчатый или если привод состоит из редуктора и открытой передачи;
- 3) выбор материалов для шестерни и колеса;
- 4) определение допускаемых напряжений;
- 5) определение крутящих моментов на шестерне и колесе рассчитываемой ступени;

- 6) проектировочный расчет передачи;
- 7) уточнение размеров передачи;
- 8) проверочный расчет на выносливость по контактным напряжениям (закрытые передачи);
- 9) проверочный расчет по напряжениям изгиба;
- 10) проверка прочности зубьев при перегрузках (при действии пиковых нагрузок).

В двухступенчатых цилиндрических редукторах в указанном порядке производится расчет одной ступени, чаще тихоходной. Затем в соответствии с ГОСТ 2185-66 выбираются размеры второй ступени и производится проверочный расчет.

Если задан коническо-цилиндрический редуктор, то первой рассчитывают коническую ступень, а затем цилиндрическую.

Если привод состоит из закрытой передачи (редуктора) и открытой цилиндрической передачи, то вначале рассчитывают закрытую передачу, а затем – открытую; открытая зубчатая пара рассчитывается на выносливость по напряжениям изгиба в указанном выше порядке.

После расчета передачи студенты переходят к расчету валов и подбору подшипников.

1.2 Выбор электродвигателя

В проектных заданиях по курсу деталей машин разрабатываются, как правило, приводы к машинам, работающим с малоизменяющейся нагрузкой, поэтому электродвигатели можно подбирать непосредственно по каталогу без проверки на нагрев. Требуемую номинальную мощность P на тихоходном валу определяют по проектной нагрузке. Например, для конвейера при заданном тяговом усилии F и скорости ленты v , при установившемся режиме P находят по формуле

$$P = \frac{F \cdot v}{1000}, \text{ кВт.} \quad (1.1)$$

Если известен крутящий момент T на приводном валу и угловая скорость его ω , рад/с (n , об/мин), то требуемая номинальная мощность вычисляют по формуле

$$P = \frac{T \cdot \omega}{1000}, \text{ кВт} \quad \text{или} \quad P = \frac{T \cdot n}{9550}, \text{ кВт,} \quad (1.2)$$

где T – в Нм.

Требуемая номинальная мощность двигателя $P_{\text{дв.}}$ определяют по формуле

$$P_{\text{дв}} = \frac{P}{\eta}, \quad (1.3)$$

где η - общий коэффициент полезного действия привода, равный произведению

частных КПД:

$$\eta = \eta_1 \cdot \eta_2 \cdot \eta_3 \dots \eta_n. \quad (1.4)$$

Значения КПД для различных типов передач приведены в табл. 1.1.

После того, как найдена требуемая номинальная мощность электродвигателя $P_{\text{дв}}$, выбирают тип и марку двигателя, для чего можно воспользоваться рекомендациями [2].

Таблица 1.1

Приближенные значения КПД передач с учетом потерь в подшипниках качения

Наименование	Закрытые передачи при жидкой смазке		Открытые передачи
	6-й и 7-й степени точности	8-й и 9-й степени точности	
Одна ступень зубчатого редуктора цилиндрическая	0,99...0,98	0,975...0,97	0,96...0,95
Одна ступень зубчатого редуктора коническая	0,98...0,96	0,96...0,95	0,95...0,94
Одна ступень червячного редуктора	При числе заходов червяка		
	1	2	3
	0,7...0,75	0,75...0,82	0,87...0,92
Цепная передача	Закрытая		Открытая
	0,96...0,98		0,92...0,94
Ременная передача	0,96-0,97		
Подшипники качения (одна пара)	0,99		
Муфта соединительная	0,98		

В курсовых проектах по деталям машин, как правило, следует выбирать трехфазные асинхронные электродвигатели серии АИР. Принятый ряд мощностей соответствует ГОСТ 13267-73. Габаритные, установочные и присоединительные размеры регламентированы ГОСТ 18709-73.

Трехфазные асинхронные электродвигатели общего применения имеют следующее обозначения:

АИ – вид двигателя (асинхронный) новой серии стран Интер-электро;
– третья буква – вариант привязки мощностей и установочных размеров:

P – привязка по I варианту,

C – привязка по II варианту:

I вариант – привязка в соответствии с *PC-3031-71* для асинхронных двигателей с высотами оси вращения 45-355 мм и степенями защиты *IP44* (*IP54*) и (*IP23*), предназначенных для внутренних поставок и поставок на экспорт;

II вариант – привязка в соответствии с нормами *CENELEK-DOKUMENT 28/64* для асинхронных двигателей с высотами оси вращения 56-315 мм и степенью защиты *IP44* (*IP54*), предназначенных только для поставок на экспорт.

В обозначение марки двигателя входят также цифры, первая из которых относится к высоте оси вращения, мм (две или три цифры); установочный размер по длине станины: буквы *S*, *M* или *L* (меньший, средний или больший); число полюсов (одна или две цифры); климатическое исполнение и категория размещения.

1.3. Допускаемые напряжения

1.3.1. Допускаемые контактные напряжения при расчете на выносливость

Стали, рекомендуемые для зубчатых колес, виды их термообработки и механические характеристики приведены в табл. 2.2, часть 1.

Допускаемые контактные напряжения определяют по формуле

$$[\sigma_H] = \frac{\sigma_{Hlimb}}{S_H} K_{HL}, \quad (1.5)$$

где σ_{Hlimb} – базовый предел контактной выносливости поверхности зубьев, соответствующих базовому числу циклов перемены напряжений N_{HO} ; S_H – коэффициент безопасности; K_{HL} – коэффициент долговечности.

Предел контактной выносливости – контактная прочность, а следовательно, базовый предел контактной выносливости σ_{Hlimb} и базовое число цик-

лов N_{HO} определяется в основном твердостью рабочих поверхностей зубьев (см. табл.1.2 и в части 1 табл. 2.3).

Коэффициент безопасности рекомендуется принимать: $S_H = 1,1$ – при нормализации, улучшении или объемной закалке зубьев (однородная структура по объему); $S_H = 1,2$ – при поверхностной закалке, цементации, азотировании (неоднородная структура по объему).

Коэффициент долговечности K_{HL} учитывает влияние срока службы и режима нагрузки передачи и определяют по формуле

$$K_{HL} = \sqrt[6]{\frac{N_{HO}}{N_{HE}}}, \quad (1.6)$$

где N_{HO} – базовое число циклов перемены напряжений, соответствующее длительному пределу выносливости (см. часть 1, табл. 2.3); N_{HE} – эквивалентное число циклов перемены напряжений.

Таблица 1.2

Приближенные значения пределов контактной выносливости

Способ термической или химико-термической обработки стали	Твердость поверхности зубьев	Группа стали	σ_{Hlimb} , МПа
Нормализация или улучшение	$HB_{cp} \leq 350$	Углеродистые или легированные	$2HB_{cp} + 70$
Объемная закалка	$HRC_{cp} 38...50$		$18 HRC_{cp} + 150$
Закалка ТВЧ	$HRC_{cp} 40...56$		$17 HRC_{cp} + 200$
Цементация	$HRC_{cp} 54...64$	Легированные	$23HRC_{cp}$
Азотирование	$HV_{cp} 550...750$		$1,5HV_{cp}$
Примечание: в расчетах принимают средние значения твердости в пределах допускаемого отклонения.			

При постоянном режиме нагрузки

$$N_{HE} = 60nct, \quad (1.7)$$

где n – частота вращения того из колес, по материалу которого определяют допускаемые напряжения, об/мин; c – число зацеплений зуба за один оборот колеса (равно числу колес, находящихся в зацеплении с рассчитываемым); t – число часов работы передачи за расчетный срок службы.

При переменных режимах нагрузки (см., например, циклограмму, изображенную на рис. 1.1).

$$N_{HE} = 60c \sum \left(\frac{T_i}{T_1}\right)^3 n_i t_i, \quad (1.8)$$

где T_i – момент, соответствующий i -ой ступени циклограммы нагружений; n_i , t_i – соответствующие этим моментам частота вращения и время работы; T_1 – максимальный из моментов, учитываемых при расчете на выносливость.

Предельные значения K_{HL} ограничиваются. Для стальных колес $K_{HL} \leq 2,6$ при объемном упрочнении и $K_{HL} \leq 1,8$ при поверхностном упрочнении. При $N_{HO}/N_{HE} < 1$ коэффициент долговечности $K_{HL} = 1$.

Для прямозубых передач, а также для косозубых передач с небольшой разностью твердости зубьев шестерни и колеса за расчетное принимается меньшее из двух допускаемых напряжений, определяемых по материалу шестерни $[\sigma_H]_1$ и колеса $[\sigma_H]_2$.

Для косозубых передач, у которых твердость зубьев шестерни значительно выше твердости зубьев колеса (например, шестерня $HB \geq 400$, колесо $HB \leq 320$), за расчетное принимают $[\sigma_H] = 0,45([\sigma_H]_1 + [\sigma_H]_2) \geq [\sigma_H]_{min}$, но не более $1,25[\sigma_H]_{min}$ (меньшего из двух значений).

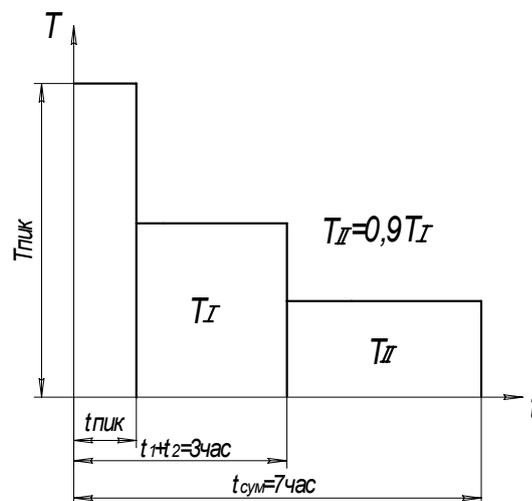


Рис.1.1. График сменной нагрузки

Пример

График сменной нагрузки приведён на рис.1.1. Срок службы передачи – $T_{сл} = 10$ лет, рабочих дней в году $n_{рд} = 240$, работа в три смены $n_{см} = 3$, частота вращения шестерни 119 об/мин, частота вращения колеса 47,6 об/мин.

При переменном режиме нагрузки число циклов равно

$$N_{HE} = 60 \cdot n_i \cdot c \cdot n_{cm} \cdot n_{рд} \cdot T_{сл} \cdot \left[\left(\frac{T_1}{T_1} \right)^3 \cdot t_1 + \left(\frac{0,9T_1}{T_1} \right)^3 \cdot t_2 \right];$$

для шестерни $N_{HE1} = 60 \cdot 1 \cdot 119 \cdot 3 \cdot 240 \cdot 10 \cdot (1 \cdot 3 + 0,9^3 \cdot 4) = 3,04 \cdot 10^8$;

для колеса $N_{HE2} = 60 \cdot 1 \cdot 47,6 \cdot 3 \cdot 240 \cdot 10 \cdot (1 \cdot 3 + 0,9^3 \cdot 4) = 1,22 \cdot 10^8$.

1.2.2. Допускаемые напряжения изгиба при расчете на выносливость

Допускаемые напряжения изгиба при расчете на выносливость определяются по формуле

$$[\sigma_F] = \frac{\sigma_{Flimb}}{S_F} K_{FC} \cdot K_{FL},$$

(1.9)

где σ_{Flimb} – базовый предел выносливости зубьев по излому от напряжений изгиба (рекомендации по выбору его, выработанные на базе экспериментальных исследований на зубчатых колесах, приведены в таблице 1.3); S_F – коэффициент безопасности (рекомендуется $S_F = 1,7 \dots 2,2$, верхнее значение для литых заготовок); K_{FC} – коэффициент, учитывающий влияние двухстороннего приложения нагрузки ($K_{FC} = 1$ – односторонняя нагрузка; $K_{FC} = 0,8 \dots 0,7$ – реверсивная нагрузка, большие значения при $HB > 350$); K_{FL} – коэффициент долговечности.

Таблица 1.3

Значения пределов выносливости зубьев по напряжениям изгиба

Способ термической или химико-термической обработки зубьев	Твердость зубьев		Группа стали	σ_{Flimb} , МПа
	поверхности	сердцевинны		
Нормализация	180 ... 350 <i>HB</i>		Углеродистая и легированные (40, 45, 50, 40X, 40XH и др.)	260+ <i>HB</i>
Улучшение				
Объемная закалка	45 ... 55 <i>HRC</i>		Легированная (40X, 40XH, 40XФА и др.)	550...600
Азотирование	55...67 <i>HRC</i>	24...40 <i>HRC</i>	Легированные (40X, 40XФА и др.)	43...19 <i>HRC</i>

Цементация	55...63 <i>HRC</i>	30...45 <i>HRC</i>	Легированные (20Х, 12ХНЗА, 25ХГТ и др.)	750...850
Примечание: в расчете принимают средние значения твёрдости в пределах допускаемого отклонения.				

При $HB \leq 350$, а также для зубчатых колес со шлифованной переходной поверхностью зубьев

$$K_{FL} = \sqrt[6]{\frac{N_{FO}}{N_{FE}}} \quad (1.10)$$

Значения K_{FL} ограничены пределами $1 \leq K_{FL} \leq 2$.

При $HB > 350$, а также для зубчатых колёс с нешлифованной переходной поверхностью зубьев

$$K_{FL} = \sqrt[9]{\frac{N_{FO}}{N_{FE}}} \quad (1.11)$$

В (1.10) и (1.11) N_{FO} – базовое число циклов перемены напряжений; $N_{FO} = 4 \cdot 10^6$ для всех сталей.

Значения K_{FL} в этом случае ограничены пределами $1 \leq K_{FL} \leq 1,6$.

При постоянном режиме нагрузки эквивалентное число циклов N_{FE} – определяют по формуле (1.7). При переменном режиме нагрузки – по аналогии с формулой (1.8)

$$N_{HE} = 60c \sum \left(\frac{T_i}{T_1}\right)^m n_i t_i \quad (1.12)$$

Показатель степени рекомендуется принимать: $m = 6$ для нормализованных и улучшенных сталей, а также при поверхностном упрочнении, если переходная поверхность шлифуется; $m = 9$ для закаленных сталей.

1.4. Распределение передаточного числа

Передаточное число привода $u_{пр}$ определяют как частное от деления угловой скорости (частоты вращения) вала двигателя $\omega_{дв}$ ($n_{дв}$) на угловую скорость (частоту вращения) рабочего органа $\omega_{р.о}$ ($n_{р.о}$):

$$u_{пр} = \frac{\omega_{дв}}{\omega_{р.о}} = \frac{n_{дв}}{n_{р.о}} \quad (1.13)$$

Вал двигателя муфтой связан с быстроходным валом редуктора; вал, на котором расположен рабочий орган, связан соединительной муфтой с тихоходным валом редуктора.

Угловая скорость рабочего органа $\omega_{p.o}$, например, барабана (механизмов подъема, лебедки или ленточного конвейера), звездочки цепного конвейера, ходового колеса механизмов передвижения, может быть определена так

$$\omega_{p.o} = \frac{2v}{D}, \text{ рад/с},$$

где v – линейная скорость цепи, ленты, каната и т.д.; D – диаметр барабана, звездочки, ходового колеса и т.п.

Если привод, кроме закрытой зубчатой передачи, включает в себя еще ременную, цепную или открытую зубчатую, то передаточное число редуктора определяют как частное от деления передаточного числа привода $u_{пр}$ на передаточное число ременной, цепной или открытой зубчатой передачи

$$u = \frac{u_{пр}}{u_{рем(ц,п)}}. \quad (1.14)$$

Значения номинальных передаточных чисел цилиндрических редукторов по ГОСТ 2185-66 приведены в табл. 1.4.

По массогабаритным показателям передачи невыгодно выполнять одноступенчатые передачи с большими передаточными отношениями. При $u \geq 8$ рекомендуют перейти к двухступенчатому, а при $u \geq 45$ – к трехступенчатому редуктору. Максимальное передаточное число трехступенчатого редуктора равно 315.

Масса и габариты редуктора зависят от того, как распределено передаточное отношение по ступеням.

Уменьшать габариты и массу редуктора можно за счёт термообработки колёс до высокой твёрдости, особенно в крупносерийном производстве. При высокой твёрдости зубьев встречаются случаи, когда главным критерием работоспособности становится прочность не по контактным, а по изгибным напряжениям.

Число ступеней можно определить по табл. 1.4, исходя из общего передаточного числа редуктора.

Таблица

1.4

Номинальные передаточные числа цилиндрических редукторов (ГОСТ 2185-66)

Тип редуктора	Номинальные передаточные числа
---------------	--------------------------------

Одноступенчатый	1 ряд	1,0	1,25	1,6	2,0	2,5	3,15
	2 ряд	1,12	1,4	1,8	2,24	2,8	3,55
	1 ряд	4,0	5,0	6,3	8,0	10,0	12,5
	2 ряд	4,5	5,6	7,1	9,0	11,2	
Двухступенчатый	1 ряд	6,3	8,0	10,0	12,5	16,0	20,0
	2 ряд	7,1	9,0	11,2	14,0	18,0	22,4
	1 ряд	25,0	31,5	40,0	50,0	63,0	-
	2 ряд	28,0	35,5	45,0	56,0	-	-
Трехступенчатый	1 ряд	31,5	40,0	50,0	63,0	80,0	100
	2 ряд	35,5	45,0	56,0	71,0	90,0	112
	1 ряд	125	160	200	250	315	400
	2 ряд	140	180	224	280	355	-

Так как тихоходная ступень нагружена больше, чем быстроходная, то передаточное отношение второй (тихоходной) ступени рекомендуют брать меньше, чем первой (быстроходной). Рекомендуется также для соосного редуктора увеличивать коэффициент ширины колёс ψ_{ba} от быстроходной к тихоходной ступени.

ГОСТ 2185-66 допускает максимальное отклонение от номинального передаточного числа для одноступенчатых редукторов: $\Delta u = \pm 2,5\%$ при $u \leq 4,5$ и $\Delta u = \pm 4\%$ при $u > 4,5$. Для двух- и трехступенчатых редукторов при всех значениях передаточного числа u отклонение не более чем на $\Delta u = \pm 4\%$.

Для распределения передаточного числа между ступенями можно пользоваться табл. 1.5-1.8.

Таблица 1.5

Передаточные числа ступеней косозубо-прямозубых
трехосных редукторов ($a_{вт}/a_{вб} = 1,4$)

Параметр	Передаточное число редуктора							
	10	11,2	12,5	14	16	18	20	22,4 и более
u_b	4,5	5		5,6	6,3		7,1	8
u_t	2,24		2,5			2,8		$u/8$

Таблица 1.6

Передаточные числа ступеней косозубо-прямозубых

соосных редукторов ($\psi_{bat} = 2,5\psi_{baб}$)

Параметр u	Передаточное число редуктора										
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5 и более
$u_б$	5		5,6		6,3		7,1	8	9		10
u_T	2	2,24		2,5		2,8			3,15		$u/u_б$

Таблица 1.7
Передаточные числа ступеней косозубых или прямозубых
трехосных редукторов ($a_{wT}/a_{wб} = 1,4$)

Параметр u	Передаточное число редуктора										
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5
$u_б$	3,55	4		4,5	5	5,6		6,3	7,1		8
u_T	2,8		3,15			3,55			4		

Таблица 1.8
Передаточные числа ступеней
косозубых или прямозубых соосных редукторов ($\psi_{bat}/\psi_{baб} = 2$)

Параметр u	Передаточное число редуктора										
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5
$u_б$	4,5		5	5,6		6,3	7,1		8	9	
u_T	2,24	2,5			2,8		3,15			3,55	

В табл. 1.5-1.8 ψ_{bat} , $\psi_{baб}$ – коэффициенты ширины соответственно тихоходной и быстроходной ступеней.

1.5. Проектный расчет на контактную выносливость

1.5.1. Межосевое расстояние

После определения вращающих моментов и частот вращения валов выполняют основной проектный расчёт зубчатых передач, который служит для предварительного определения размеров передачи.

Ориентировочное значение межосевого расстояния (a_w) определяют по формуле

$$a_w = K_a (u \pm 1) \sqrt[3]{\frac{T_2 K_{H\beta}}{u^2 \psi_{ba} [\sigma_H]^2}}, \text{ мм},$$

(1.15)

где K_a – вспомогательный коэффициент (для прямозубых передач $K_a = 495 \text{ МПа}^{1/3}$, для косозубых и шевронных передач $K_a = 430 \text{ МПа}^{1/3}$); T_2 – крутящий момент на колесе, Нм; $\psi_{ba} = \frac{b_w}{a_w}$ – коэффициент ширины колеса относительно межосевого расстояния.

Знак «+» в (1.15) соответствует внешнему зацеплению, знак «-» – внутреннему.

Одним из важных геометрических параметров передачи является также коэффициент относительной ширины шестерни ψ_{bd} , связанный с ψ_{ba} следующим образом:

$$\psi_{bd} = \frac{\psi_{ba}(u+1)}{2}.$$

(1.16)

Увеличение $\psi_{bd}(\psi_{ba})$, или относительной ширины колес, позволяет уменьшить габариты и массу передачи, но требует повышенной жесткости и точности конструкции. В противном случае появится значительная неравномерность распределения нагрузки по ширине зубчатого венца.

Значения коэффициента ψ_{ba} установлены ГОСТ 2185-66: 0,100; 0,125; 0,160; 0,200; 0,250; 0,315; 0,400; 0,500; 0,630; 0,800; 1,0; 1,25.

Рекомендуется применять следующие значения коэффициента ширины: $\psi_{ba} = 0,25 \dots 0,5$ – для прямозубых редукторов общего назначения, $\psi_{ba} = 0,2 \dots 0,4$ – для косозубых редукторов общего назначения, $\psi_{ba} = 0,5 \dots 0,63$ – для редукторов с шевронными колёсами; $\psi_{ba} \leq 0,25$ – для раздвоенных ступеней шевронных редукторов и быстроходных ступеней соосных редукторов. При выборе коэффициента ψ_{bd} можно пользоваться рекомендациями табл. 1.9.

Таблица 1.9

Рекомендуемые значения коэффициентов ширины

Расположение колеса относительно опор	Рекомендуемые значения	Твердость рабочих поверхностей зубьев	
		$HB_2 \leq 350$ или HB_1 и $HB_2 \leq 350$	HB_1 и $HB_2 > 350$
Симметричное	ψ_{ba}	0,25...0,5	0,25...0,315
	ψ_{bd}	0,8...1,6	0,4...0,9

Несимметричное	ψ_{ba}	0,25...0,4	0,2...0,25
	ψ_{bd}	0,6...1,2	0,3...0,65
Консольное	ψ_{ba}	0,2...0,25	0,15...0,2
	ψ_{bd}	0,3...0,7	0,2...0,5
Примечание: для шевронных передач ψ_{ba} увеличивается в 1,1...1,4 раза; для подвижных колес коробок передач $\psi_{ba} = 0,1...0,2$.			

Коэффициент, учитывающий неравномерность распределения нагрузки по ширине венца ($K_{H\beta}$), принимается в зависимости от ψ_{bd} по таблице 1.10.

Полученное межосевое расстояние следует округлить до ближайшего стандартного (ГОСТ 2185-66), для одноступенчатого редуктора по табл.1.11 и для двухступенчатого - по табл.1.12.

Таблица 1.10

Значение коэффициентов $K_{H\beta}$ и $K_{F\beta}$, учитывающих неравномерность распределения нагрузки по ширине венца

Относительная ширина колеса ψ_{bd}	Симметричное расположение шестерни относительно опор				Шестерня расположена несимметрично относительно опор								Консольное расположение одного из колес			
					весьма жесткий вал				менее жесткий вал							
	$K_{H\beta}$		$K_{F\beta}$		$K_{H\beta}$		$K_{F\beta}$		$K_{H\beta}$		$K_{F\beta}$		$K_{H\beta}$		$K_{F\beta}$	
	при твердости рабочих поверхностей зубьев НВ				при твердости рабочих поверхностей зубьев НВ				при твердости рабочих поверхностей зубьев НВ				при твердости рабочих поверхностей зубьев НВ			
	>350	<350	>350	<350	>350	<350	>350	<350	>350	<350	>350	<350	>350	<350	>350	<350
0,2	1,0	1,0	1,0	1,0	1,01	1,0	1,02	1,01	1,06	1,02	1,1	1,05	1,15	1,07	1,25	1,13
0,4	1,01	1,0	1,03	1,01	1,05	1,02	1,07	1,04	1,12	1,05	1,2	1,12	1,35	1,15	1,55	1,28
0,6	1,03	1,01	1,05	1,02	1,09	1,04	1,13	1,07	1,20	1,08	1,3	1,17	1,60	1,24	1,90	1,50
0,8	1,06	1,03	1,08	1,05	1,14	1,06	1,20	1,11	1,27	1,12	1,44	1,23	1,85	1,30	2,30	1,70
1,0	1,10	1,04	1,15	1,08	1,18	1,08	1,27	1,15	1,37	1,15	1,57	1,32				
1,2	1,13	1,05	1,18	1,10	1,25	1,10	1,37	1,20	1,50	1,18	1,72	1,40				
1,4	1,15	1,07	1,25	1,13	1,32	1,13	1,50	1,25	1,60	1,23	1,85	1,50				
1,6	1,20	1,08	1,30	1,16	1,40	1,16	1,60	1,32	-	1,28	-	1,60				

Таблица 1.11

Одноступенчатые редукторы, a_w , мм

1 ряд	50	63	80	100	125	160	200	250	315	400	500	630	800	1000
2 ряд	–	71	90	112	140	180	224	280	355	450	560	710	900	1120

Таблица 1.12

Двухступенчатые редукторы, a_w , мм

Быстроходная ступень a_{wb}	50	63	80	100	112	125	140	160	180
Тихоходная ступень a_{wt}	63	80	112	140	160	180	200	224	250
Быстроходная ступень a_{wb}	200	224	250	280	315	355	400	450	500
Тихоходная ступень a_{wt}	280	315	355	400	450	500	560	630	710

1.5.2. Выбор модуля и числа зубьев

Основными параметрами передачи являются межосевое расстояние a_w , ширина зубчатого венца колеса b_w (в последнее время используется обозначение b_2), модуль m и число зубьев шестерни z_1 и колеса z_2 , передаточное число u .

Значение **модуля** (для косозубых передач – нормального модуля) рекомендуется принимать в пределах $m(m_n) = (0,01 \div 0,02) a_w$.

Полученное значение модуля m округляется до стандартного (табл.1.13). Для силовых передач рекомендуется принимать $m \geq 1,5$ мм.

Затем определяют **суммарное число зубьев** $z_c = z_1 + z_2$ для прямозубых передач по формуле

$$z_c = \frac{2a_w}{m}, \quad (1.17)$$

для косозубых и шевронных

$$z_c = \frac{2a_w \cos \beta}{m_n}, \quad (1.18)$$

где β - угол наклона зубьев, рекомендуется для косозубых колес $\beta = 8 \dots 15^\circ$, для шевронных – $\beta = 25 \dots 40^\circ$.

Значения модулей $m (m_n)$ по ГОСТ 9563-60, мм

1 ряд	1,5	2	2,5	3	4	5	6	8	10	12
2 ряд	1,75	2,25	2,75	3,5	4,5	5,5	7	9	11	14

Чтобы избежать нарезания колес со смещением, значения модуля выбирают таким, чтобы z_c в формулах (1.17) и (1.18) было целым числом и $z_1 \geq z_{min}$. У косозубых и шевронных передач можно при этом варьировать значением угла β .

Число зубьев шестерни определяют по формуле

$$z_1 = \frac{z_c}{u + 1} . \quad (1.19)$$

Минимальное число зубьев для прямозубых колес составляет $z_{min} = 17$, для косозубых и шевронных – $z_{min} = 17 \cos^3 \beta$.

При числе зубьев шестерни z_1 меньше z_{min} необходимо применять передачи со смещением. Коэффициент смещения при этом определяют: для прямозубой шестерни

$$x_1 = \frac{17 - z_1}{17} , \quad (1.20)$$

для косозубой шестерни

$$x_1 = \frac{17 \cos^3 \beta - z_1}{17} . \quad (1.21)$$

Число зубьев колеса

$$z_2 = z_c - z_1 .$$

После определения z_1 и z_2 уточняют передаточное число u .

Фактическое передаточное число

$$u_{\phi} = \frac{z_2}{z_1} . \quad (1.22)$$

Отклонение от заданного числа u

$$\Delta u = \frac{u_{\phi} - u}{u_1} 100 , \% . \quad (1.23)$$

Передачи со смещением могут применяться и в случае, когда необходимо иметь определенное значение модуля m при стандартном значении межосевого расстояния a_w . Суммарное значение коэффициента смещения $x_\Sigma = x_1 + x_2$ определяется по формуле

$$x = x_1 + x_2 = \frac{(z_1 + z_2)(\text{inv}\alpha_{tw} - \text{inv}\alpha_t)}{2\text{tg}\alpha}; \quad (1.24)$$

$$\cos\alpha_{tw} = \frac{(z_1 + z_2)m_n \cos\alpha_t}{2a_w \cos\beta}; \quad (1.25)$$

$$\text{tg}\alpha_t = \frac{\text{tg}\alpha}{\cos\beta}, \quad (1.26)$$

где $\alpha_{tw} = 20^\circ$ – угол зацепления; α_t – угол профиля; α – угол профиля зуба рейки.

Значения инволют углов выбираются по таблицам, которые в данном пособии не приводятся.

При необходимости нарезать со смещением можно или только шестерню (при положительном x_Σ), или только колесо (при отрицательном x_Σ).

Диаметры начальных, делительных окружностей, окружностей вершин зубьев и впадин вычисляются по формулам, приведённым ранее (см. часть 1, стр.43).

1.5.3. Определение ширины колеса

Рабочая ширина венца зубчатого колеса определяется по формуле

$$b_w = \psi_{ba} \cdot a_w. \quad (1.27)$$

Полученное значение b_w округляется до ближайшего стандартного по ГОСТ 6636-69 ряд $R_a 40$ (см. приложение 2).

В открытых передачах ширина колеса определяется по формуле

$$b_w = \psi_{bd} \cdot d_1. \quad (1.28)$$

Ширина шестерни b_1 для компенсации неточностей сборки выполняется несколько больше, примерно на 5...10 мм ширины колеса. Но при проверочном расчете подставляется величина b_w .

1.6. Проверочный расчет на выносливость по контактным напряжениям

После уточнения размеров передачи производится проверочный расчет по формуле

$$\sigma_H = 22,4 Z_H Z_M Z_\varepsilon \frac{1}{a_w} \sqrt{\frac{T_1 (u \pm 1)^3}{b_w u}} \cdot K_{H\alpha} \cdot K_{H\beta} \cdot K_{H\nu} \leq [\sigma_H]. \quad (1.29)$$

Величина расчетных контактных напряжений одинакова для шестерни и колеса, поэтому расчет выполняют для того из колес пары, у которого меньше допускаемое напряжение $[\sigma_H]$.

В формулу (1.29) крутящий момент T_1 подставляется в Нм, a_w и b_w в мм. Рассмотрим величины, входящие в эти формулы:

- Z_H – коэффициент, учитывающий форму сопряженных поверхностей зубьев

$$Z_H = \sqrt{\frac{2 \cos^2 \beta}{\sin 2\alpha_w}}; \quad (1.30)$$

при $x_1 = 0$ и $x_2 = 0$, $\alpha_w = \alpha = 20^\circ$, следовательно, $Z_H = 1,77 \cos \beta$; для прямозубой передачи $\beta = 0$ и $Z_H = 1,77$;

- Z_M – коэффициент, учитывающий механические свойства материалов сопряженных зубчатых колес; для стальных колес $Z_M = 275 \text{ МПа}^{1/2}$;

- Z_ε – коэффициент, учитывающий суммарную длину контактных линий, для прямозубых передач

$$Z_\varepsilon = \sqrt{\frac{4 - \varepsilon_\alpha}{3}}; \quad (1.31)$$

для косозубых и шевронных передач при $\varepsilon_\beta \geq 0,9$

$$Z_\varepsilon = \sqrt{\frac{1}{\varepsilon_\alpha}}; \quad (1.32)$$

$\varepsilon_\beta = \frac{b_w \sin \beta}{\pi m}$ – коэффициент осевого перекрытия; ε_α – коэффициент торцевого перекрытия; для передач без смещения

$$\varepsilon_\alpha = [1,88 - 3,2(\frac{1}{z_1} \pm \frac{1}{z_2})] \cos \beta, \quad (1.33)$$

знак «+» в (1.33) соответствует внешнему зацеплению, знак «-» – внутреннему (для прямозубых передач рекомендуется $\varepsilon_\alpha \geq 1,2$, для косозубых – $\varepsilon_\alpha \geq 1$);

- $K_{H\alpha}$ – коэффициент, учитывающий распределение нагрузки между зубьями; для прямозубых передач $K_{H\alpha} = 1$, для косозубых и шевронных $K_{H\alpha}$ выбирается по таблице 1.14;

- $K_{H\beta}$ – коэффициент, учитывающий распределение нагрузки по ширине венца (табл. 1.10);

- K_{Hv} – коэффициент динамической нагрузки, который выбирается в зависимости от степени точности передачи, твёрдости поверхности зубьев и окружной скорости (табл.1.16).

Таблица 1.14

Значения коэффициента $K_{H\alpha}$

Окружная скорость v , м/с	Степень точности			
	6	7	8	9
2,5	1,01	1,03	1,05	1,13
5	1,02	1,05	1,09	1,16
10	1,03	1,07	1,13	-
15	1,04	1,09	-	-
20	1,05	1,12	-	-
25	1,06	-	-	-

Вычисляют окружную скорость

$$v = \frac{\omega_1 d_{w1}}{2 \cdot 1000} = \frac{\pi d_{w1} n_1}{60 \cdot 1000}, \text{ м/с}. \quad (1.34)$$

По окружной скорости, пользуясь таблицей 1.15, выбирают степень точности передачи.

1.7. Проверочный расчет на выносливость по напряжениям изгиба

Расчет выполняют по тому из зубчатых колес пары, у которого **меньше** отношение $\frac{[\sigma_F]}{Y_F}$.

Проверку зубьев на выносливость при изгибе выполняют по формуле

$$\sigma_F = Y_F Y_\beta Y_\varepsilon \frac{2000 T_1 K_{F\alpha} K_{F\beta} K_{Fv}}{b_w d_{w1} m(m_n)} \leq [\sigma_F]. \quad (1.35)$$

Рекомендации по выбору степени точности цилиндрических передач

Степень точности передач	Окружная скорость колес v , м/с	
	прямозубых	Косозубых
6	до 15	до 30
7	до 10	до 15
8	до 6	до 10
9	до 2	до 4

Здесь крутящий момент T_1 в Нм; d_w , b_w и m (m_n) в мм.

- Y_F – коэффициент формы зуба, определяется по табл. 1.17, для эквивалентного числа зубьев;
- Y_β – коэффициент, учитывающий наклон зубьев; для прямозубых передач $Y_\beta = 1$; для косозубых и шевронных $Y_\beta = 1 - \frac{\beta}{140^\circ}$ и при $\beta > 42^\circ$ $Y_\beta = 0,7$;
- Y_ϵ – коэффициент, учитывающий перекрытие зубьев; ориентировочно можно принимать $Y_\epsilon = 1$;
- $K_{F\alpha}$ – коэффициент, учитывающий распределение нагрузки между зубьями; для прямозубых передач и косозубых с $\epsilon_\beta \leq 1$ принимают $K_{F\alpha} = 1$, для косозубых с $\epsilon_\beta > 1$ и шевронных передач

$$K_{F\alpha} = \frac{4 + (\epsilon_\alpha - 1)(n - 5)}{4\epsilon_\alpha}; \quad (1.36)$$

где n – степень точности передачи;

- $K_{F\beta}$ – коэффициент, учитывающий распределение нагрузки по ширине венца, выбирается по табл. 1.10;
- K_{FV} – коэффициент динамической нагрузки (табл. 1.16).

1.8. Проверочный расчёт прочности зубьев при перегрузках

Максимальные моменты (например, момент T_{max} на рис. 1.1.), неучтенные при расчете на выносливость, могут привести к статическому разрушению зубьев. Поэтому после определения размеров передачи по условию выносливости необходимо проверить на статическую прочность при перегрузках, если они заданы.

Для предотвращения остаточных деформаций или хрупкого разрушения поверхностного слоя контактное напряжение σ_{Hmax} не должно превышать допускаемое напряжение $[\sigma_H]_{max}$.

Таблица 1.16

Значения коэффициентов K_{HV} и K_{FV}

Степень точности	Твёрдость зубьев	Коэффициенты	Скорость V , м/с					
			1	2	4	6	8	10
6	а	K_{HV}	1,03	1,06	1,12	1,17	1,23	1,28
			1,01	1,02	1,03	1,04	1,06	1,07
		K_{FV}	1,06	1,05	1,10	1,15	1,20	1,25
			1,02	1,04	1,07	1,10	1,15	1,18
	б	K_{HV}	1,02	1,04	1,07	1,10	1,15	1,18
			1,00	1,00	1,02	1,02	1,03	1,04
K_{FV}	1,02	1,04	1,08	1,11	1,14	1,17		
	1,01	1,02	1,03	1,04	1,06	1,07		
7	а	K_{HV}	1,04	1,07	1,14	1,21	1,29	1,36
			1,02	1,03	1,05	1,06	1,07	1,08
		K_{FV}	1,08	1,16	1,33	1,50	1,67	1,80
			1,03	1,06	1,11	1,16	1,22	1,27
	б	K_{HV}	1,03	1,05	1,09	1,14	1,19	1,24
			1,00	1,01	1,02	1,03	1,03	1,04
K_{FV}	1,03	1,05	1,09	1,13	1,17	1,22		
	1,01	1,02	1,03	1,05	1,07	1,08		
8	а	K_{HV}	1,04	1,08	1,16	1,24	1,32	1,40
			1,01	1,02	1,04	1,06	1,07	1,08
		K_{FV}	1,10	1,20	1,38	1,58	1,78	1,96
			1,03	1,06	1,11	1,17	1,23	1,29
	б	K_{HV}	1,03	1,06	1,10	1,16	1,22	1,26
			1,01	1,01	1,02	1,03	1,04	1,05
K_{FV}	1,04	1,06	1,12	1,16	1,21	1,26		
	1,01	1,02	1,03	1,05	1,07	1,08		
9	а	K_{HV}	1,05	1,10	1,20	1,30	1,40	1,50
			1,01	1,03	1,05	1,07	1,09	1,12
		K_{FV}	1,13	1,28	1,50	1,77	1,98	2,25
			1,04	1,07	1,14	1,21	1,28	1,35
	б	K_{HV}	1,04	1,07	1,14	1,21	1,28	1,35
			1,01	1,01	1,02	1,03	1,04	1,05
K_{FV}	1,04	1,07	1,14	1,21	1,27	1,34		
	1,01	1,02	1,04	1,06	1,08	1,09		

Примечания: 1) твёрдость поверхностей зубьев: а) $HB_1 \leq 350$, $HB_2 \leq 350$; $HRC_1 \geq 45$, $HB_2 \leq 350$; б) $HRC_1 > 45$, $HRC_2 > 45$;

2) верхние цифры указаны для прямозубых, нижние- для косозубых колёс.

Таблица 1.17

Значения коэффициента формы зуба Y_F при коэффициенте смещения $x=0$

Эквивалентное число зубьев, z_3	Y_F								
17	4,26	211	4,01	28	3,82	40	3,70	80	3,61
18	4,20	22	4,00	30	3,80	45	3,68	100	3,60
19	4,11	24	3,92	32	3,78	50	3,65	150	3,60
20	4,08	25	3,90	37	3,71	60	3,62	рейка	3,63

Примечание: z_3 – эквивалентное число зубьев, которое определяется для цилиндрического прямозубого колеса $z_3=z$; для цилиндрического косозубого колеса $z_3=z/\cos^3\beta$; для конического прямозубого колеса $z_3=z/\cos\delta$; для конического с круговым зубом $z_3=z/(\cos\delta\cos\beta_m)$.

Максимальные контактные напряжения σ_{Hmax} при действии момента T_{max} можно определить по формуле

$$\sigma_{Hmax} = \sigma_H \sqrt{\frac{T_{max} P_{пасп}}{T P_{дв}}} \leq [\sigma_H]_{max}, \quad (1.37)$$

где σ_H – контактные напряжения по (1.29); T – номинальный момент, соответствующий нормально протекающему рабочему процессу; $P_{пасп}$ – мощность двигателя, указанная в его технических данных; $[\sigma_H]_{max}$ – предельное допустимое контактное напряжение ($[\sigma_H]_{max} = 2,8\sigma_T$, где σ_T – предел текучести материала, – при нормализации, улучшении или объемной закалке зубьев; $[\sigma_H]_{max} = 44\text{HRC}$ при цементации зубьев и закалке т.в.ч.; $[\sigma_H]_{max} \approx 35\text{HRC}$ при азотировании зубьев).

Максимальные напряжения изгиба определяются по формуле

$$[\sigma_F]_{max} = \sigma_F \frac{T_{max} P_{пасп}}{T P_{дв}} \leq [\sigma_F]_{max}, \quad (1.38)$$

где σ_F – напряжения изгиба по (1.35); $[\sigma_F]_{max}$ – предельное допустимое напряжение.

Предельно допускаемое напряжение изгиба [2]

$$[\sigma_{Fmax}] = \frac{\sigma_{Flimb} \cdot Y_{Nmax} \cdot k_{ST}}{S_{ST}}, \quad (1.39)$$

где $Y_{N \max}$ – максимально возможное значение коэффициента долговечности ($Y_{N \max} = 4$ для сталей с объёмной термообработкой; $Y_{N \max} = 2,5$ для сталей с поверхностной термообработкой); k_{ST} – коэффициент влияния частоты приложения пиковой нагрузки (в случае единичных перегрузок $k_{ST} = 1,2 \dots 1,3$, бóльшие значения – для объёмной термообработки; при многократном (порядка 1000) действии перегрузок $k_{ST} = 1$); S_{ST} – коэффициент запаса прочности (обычно $S_{ST} = 1,75$) [2].

Пример.

Проверочный расчет передачи на максимальные нагрузки

Ранее рассчитаны для двигателя $P_{\text{пасп}} = 4$ кВт, $P_{\text{дв}} = 3,84$ кВт, допускается кратковременная (пиковая) перегрузка $T_{\text{max}} / T = 2,2$. Определенные по формулам (1.29) и (1.35) напряжения $\sigma_H = 387$ МПа и $\sigma_F = 72,8$ МПа.

Определяем максимальные контактные напряжения от действия пиковой нагрузки

$$\sigma_{H \max} = \sigma_H \sqrt{\left(\frac{T_{\text{max}}}{T}\right) \cdot \left(\frac{P_{\text{дв.пасп}}}{P_{\text{дв.треб}}}\right)} = 387 \sqrt{2,2 \cdot \frac{4}{3,84}} = 586 \text{ МПа} .$$

Вычисляем предельно допускаемые контактные напряжения для колеса

$$[\sigma_{H \max}]_2 = 2,8 \cdot 340 = 952 \text{ МПа} .$$

Условия прочности выполняются, т. к. $\sigma_{H \max} < [\sigma_{H \max}]_2$, $586 < 952$ МПа.

Определяем максимальные напряжения изгиба от действия пиковой нагрузки

$$\sigma_{F \max 1} = \sigma_{F1} \cdot \frac{T_{\text{max}}}{T} \cdot \frac{P_{\text{дв.пасп}}}{P_{\text{дв.треб}}} = 72,8 \cdot 2,2 \cdot \frac{4}{3,84} = 167 \text{ МПа} .$$

Вычисляем предельно допускаемые напряжения изгиба для шестерни

$$[\sigma_{F \max}]_1 = \frac{476 \cdot 4 \cdot 1,3}{2} = 1238 \text{ МПа} .$$

Условия выполняются, т. к. $\sigma_{F \max 1} < [\sigma_{F \max}]_1$, $167 < 1238$ МПа.

В соответствии с расчетами передача всем условиям удовлетворяет.

1.9. Особенности расчёта открытых цилиндрических зубчатых передач

Расчёт открытой цилиндрической передачи производится на выносливость по напряжениям изгиба.

При проектном расчёте предварительно выбирают ψ_{bd} – коэффициент ширины колеса по табл. 1.9. С целью уменьшения шума в быстроходных передачах число зубьев шестерни z_1 рекомендуется принимать больше 25, а для тихоходных передач можно принимать меньше.

Расчёт производят шестерни по формуле

$$m = K_m \sqrt[3]{\frac{T_2 \cdot K_{F\beta} \cdot Y_F}{z_1^2 \psi_{bd} [\sigma_F]_1}} ; \quad (1.40)$$

где K_m – вспомогательный коэффициент, для прямозубых передач $K_m = 14$, для косозубых и шевронных $K_m = 11,2$.

Коэффициенты $K_{F\beta}$ и Y_{F1} выбираются так же как и для закрытых передач по табл.1.10 и табл.1.17.

Полученное значение модуля m округляют до ближайшего стандартного по ГОСТ 9563-60 (см.табл.1.13). Затем уточняют все остальные параметры передачи и производят проверочный расчёт по контактным напряжениям и по напряжениям изгиба.

2. РАСЧЕТ ЗАКРЫТОЙ КОНИЧЕСКОЙ ЗУБЧАТОЙ ПЕРЕДАЧИ

2.1. Проектный расчет

2.1.1. Внешний делительный диаметр колеса

Определяется главный параметр - **внешний делительный диаметр** колеса d_{e2} , мм [5]:

$$d_{e2} \geq 1650 \sqrt[3]{u \cdot T_2 \cdot K_{H\beta} / (\vartheta_H [\sigma_H]^2)}, \quad (2.1)$$

где T_2 – крутящий момент на колесе, Н·м; $K_{H\beta}$ – коэффициент, учитывающий распределение нагрузки по ширине венца (для прирабатываемых колес с прямыми зубьями $K_{H\beta} = 1$, с круговыми зубьями $K_{H\beta} = 1,1$); ϑ_H – коэффициент вида конических колес (для прямозубых колес $\vartheta_H = 1$; для колес с круговыми зубьями $\vartheta_H = 1,85$ – при твердости колеса и шестерни $\leq 350 HB$ и $\vartheta_H = 1,5$ – при твердости колеса $\leq 350 HB$ и шестерни $\geq 45 HRC$).

Допускаемое контактное напряжение $[\sigma_H]$ определяется так же, как и для цилиндрических передач.

Полученное значение внешнего делительного диаметра колеса d_{e2} для нестандартных передач округлить до ближайшего значения из ряда нормальных линейных размеров (приложение 1).

2.1.2. Внешнее конусное расстояние и углы делительных конусов

Внешнее конусное расстояние (R_e) определяется по формуле

$$R_e = d_{e2} / (2 \sin \delta_2), \text{ мм.} \quad (2.2)$$

Значение R_e до целого числа не округлять.

Углы делительных конусов шестерни δ_1 и колеса δ_2 вычисляются по формулам

$$\delta_2 = \arctg u; \quad \delta_1 = 90 - \delta_2. \quad (2.3)$$

Точность вычислений до четвёртого знака после запятой [2].

2.1.3. Ширина колёс и модуль

Определяется **ширина** зубчатого венца шестерни и колес b , мм,

$$b = \psi_R R_e, \quad (2.4)$$

где $\psi_R = 0,285$ – коэффициент ширины венца. Значение b необходимо округлять до целого числа по ряду $Ra 40$ (см. приложение 1).

Внешний окружной модуль m_e – для прямозубых колес, m_{te} – для колес с круговыми зубьями вычисляют по формуле

$$m_e(m_{te}) = 14T_2 \cdot 10^3 K_{F\beta} / (\vartheta_F d_{e2} b [\sigma_F]), \text{ мм}, \quad (2.5)$$

где $K_{F\beta}$ - коэффициент, учитывающий распределение нагрузки по ширине венца (для прирабатывающихся колес с прямыми зубьями $K_{F\beta} = 1$, с круговыми зубьями $K_{F\beta} = 1,08$); ϑ_F – коэффициент вида конических колёс (для прямозубых колес $\vartheta_F = 0,85$; для колес с круговыми зубьями $\vartheta_F = 1$).

Значение модуля, полученное с точностью до двух знаков после запятой, до стандартной величины не округлять. В силовых конических передачах следует принимать $m_e (m_{te}) \geq 1,5$ мм, при этом в открытых передачах значение модуля m_e увеличить на 30% из-за повышенного изнашивания зубьев.

2.1.4. Числа зубьев колёс

Числа зубьев колеса z_2 , и шестерни z_1 определяют по формулам

$$z_2 = d_{e2} / m_e(m_{te}); \quad (2.6)$$

$$z_1 = z_2 / u. \quad (2.7)$$

Полученные значения z_1 и z_2 округлить в ближайшую сторону до целого числа. Из условия уменьшения шума и отсутствия подрезания зубьев рекомендуется принять $z_1 \geq 15$ – для колес с круговыми зубьями, $z_1 \geq 18$ – для прямозубых колес.

После вычисления чисел зубьев уточняют фактическое передаточное число u_ϕ

$$u_\phi = z_2 / z_1, \quad (2.8)$$

и определяют его **отклонение** Δu от заданного

$$\Delta u = |u_{\phi} - u| / u \cdot 100\% \leq 4\%. \quad (2.9)$$

При невыполнении нормы отклонения передаточного числа Δu следует изменить z_1 и z_2 .

Определяют **действительные углы** делительных конусов шестерни δ_1 и колеса δ_2 :

$$\delta_2 = \arctg u_{\phi}; \quad (2.10)$$

$$\delta_1 = 90 - \delta_2. \quad (2.11)$$

2.1.5. Выбор коэффициентов смещения

Для конических передач с разностью средних твердостей шестерни и колеса $HB_{1cp} - HB_{2cp} \leq 100$ выбрать из табл. 2.1 коэффициент смещения инструмента x_{e1} для прямозубой шестерни и x_{n1} для шестерни с круговым зубом. Коэффициенты смещения колес соответственно $x_{e2} = -x_{e1}$ и $x_{n2} = -x_{n1}$. Если $HB_{1cp} > 100$ и $HB_{2cp} > 100$, то $x_1 = x_2 = 0$.

Таблица 2.1

Коэффициенты смещения x_{e1} и x_{n1} для шестерён конических передач

z_1	x_{e1} при передаточном числе u					x_{n1} при передаточном числе u				
	2,0	2,5	3,15	4,0	5,0	2,0	2,5	3,15	4,0	5,0
12	-	0,50	0,53	0,56	0,57	0,32	0,37	0,39	0,41	0,42
13	0,44	0,48	0,52	0,54	0,55	0,30	0,35	0,37	0,39	0,40
14	0,42	0,47	0,50	0,52	0,53	0,29	0,33	0,35	0,37	0,38
15	0,40	0,45	0,48	0,50	0,51	0,27	0,31	0,33	0,35	0,36
16	0,38	0,43	0,46	0,48	0,49	0,26	0,30	0,32	0,34	0,35
18	0,36	0,40	0,43	0,45	0,46	0,24	0,27	0,30	0,32	0,32
20	0,34	0,37	0,40	0,42	0,43	0,22	0,26	0,28	0,29	0,29
25	0,29	0,33	0,36	0,38	0,39	0,19	0,21	0,24	0,25	0,25
30	0,25	0,28	0,31	0,33	0,34	0,16	0,18	0,21	0,22	0,22
40	0,20	0,22	0,24	0,26	0,27	0,11	0,14	0,16	0,17	0,17

Для передач, у которых z_1 и u отличаются от указанных в табл. 2.1, коэффициенты x_{e1} и x_{n1} принимают с округлением в большую сторону.

Внешние диаметры шестерни и колеса вычисляют по формулам, приведенным ранее (см. часть 1, стр. 53). Точность вычисления делительных диаметров колес до 0,01 мм.

Средние делительные диаметры шестерни d_1 и колеса d_2 определяются по формулам:

$$d_1 \approx 0.857d_{e1}; \quad (2.12)$$

$$d_2 \approx 0.857d_{e2}. \quad (2.13)$$

Значения d_1 и d_2 до целого числа не округлять.

2.2. Проверочный расчет

2.2.1. Проверочный расчет по контактным напряжениям

Пригодность заготовок колес. Для конических шестерни и колеса вычисляют размеры заготовки, мм: $D_{\text{заг}} = d_{ae} + 6$ мм; $S_{\text{заг}} = 8m_e(m_{te})$. Полученные расчётом значения $D_{\text{заг}}$ и $S_{\text{заг}}$ сравнивают с предельными размерами $D_{\text{пред}}$ и $S_{\text{пред}}$. Условие пригодности заготовок колес: $D_{\text{заг}} \leq D_{\text{пред}}$; $S_{\text{заг}} \leq S_{\text{пред}}$. При невыполнении неравенств изменить материал колес или вид термической обработки.

После уточнения размеров передачи производят **проверочный расчет по контактным напряжениям** по формуле

$$\sigma_H = 470 \sqrt{\frac{F_t \sqrt{u_\phi + 1}}{\vartheta_H d_{e2} b} \cdot K_{H\alpha} \cdot K_{H\beta} \cdot K_{H\nu}} \leq [\sigma_H], \quad (2.14)$$

где $F_t = 2T_2 \cdot 10^3 / d_2$ – окружная сила в зацеплении, Н; $K_{H\alpha}$ – коэффициент, учитывающий распределение нагрузки между зубьями колес (принимают $K_{H\alpha} = 1$); $K_{H\nu}$ – коэффициент динамической нагрузки (определяют в зависимости от окружной скорости колес $u = \omega_2 d_{m2} / (2 \cdot 10^3)$, м/с, и степени точности передачи, см. табл. 1.10); ω_2 – угловая скорость вала колеса редуктора или открытой передачи, 1/с.

Допускаемая недогрузка передачи ($\sigma_H < [\sigma_H]$) не более 10%, и перегрузка ($\sigma_H > [\sigma_H]$) до 5%. Если условие прочности не выполняется, то следует изменить ширину венца колеса и шестерни b . Если эта мера не даст должного результата, то либо надо увеличить внешний делительный диаметр колеса d_{e2} , либо назначить другие материалы колес или другую термообработку, пересчитать допускаемые контактные напряжения и повторить весь расчет передачи.

2.2.2. Проверочный расчёт по напряжениям изгиба

Напряжения изгиба зубьев шестерни σ_{F1} и колеса σ_{F2} , МПа:

$$\sigma_{F2} = Y_{F2} Y_{\beta} \frac{F_t K_{F\alpha} K_{F\beta} K_{Fv}}{\vartheta_F b m_e (m_{te})} \leq [\sigma_F]; \quad (2.15)$$

$$\sigma_{F1} = \sigma_{F2} \frac{Y_{F1}}{Y_{F2}} \leq [\sigma_F]. \quad (2.16)$$

где $K_{F\alpha}$ – коэффициент, учитывающий распределение нагрузки между зубьями колес (принимают $K_{F\alpha} = 1$); K_{Fv} – коэффициент динамической нагрузки, определяется аналогично коэффициенту K_{Hv} (см. табл. 1.10); Y_{F1} и Y_{F2} – коэффициенты формы зуба шестерни и колеса, определяют по табл. 2.2 в зависимости от эквивалентного числа зубьев шестерни z_{v1} и колеса z_{v2} по формулам:

для прямозубых колес

$$z_{v1} = z_1 / \cos \delta_1; \quad (2.17)$$

$$z_{v2} = z_2 / \cos \delta_2,$$

для колес с круговыми зубьями

$$z_{v1} = z_1 / (\cos \delta_1 \cos^3 \beta); \quad (2.18)$$

$$z_{v2} = z_2 / (\cos \delta_2 \cos^3 \beta),$$

$\beta = 35^\circ$ – угол наклона зубьев; Y_{β} – коэффициент, учитывающий наклон зуба ($Y_{\beta} = 1 - \beta/140^\circ$; для $\beta = 35^\circ$ $Y_{\beta} = 1 - 35^\circ/140^\circ = 0,75$).

Допускаемые напряжения изгиба шестерни и колеса $[\sigma]_{F1}$ и $[\sigma]_{F2}$ определяются так же, как для цилиндрических передач.

Если при проверочном расчете σ_F значительно меньше $[\sigma]_F$, то это допустимо, так как нагрузочная способность большинства зубчатых передач ограничивается контактной прочностью. Если σ_F превышает $[\sigma]_F$ более чем на 5%, то надо увеличить модуль m_e (m_{te}), соответственно пересчитать числа зубьев шестерни и колеса и повторить проверочный расчет на изгиб. При этом внешний делительный диаметр колеса d_{e2} не изменяется, а следовательно, не нарушается контактная прочность передачи.

Таблица 2.2

Значения коэффициента формы зуба Y_F при коэффициенте смещения x

z_v	Коэффициент смещения режущего инструмента x										
	-0,5	0,4	-0,3	-0,2	-0,1	0	+0,1	+0,2	+0,3	+0,4	+0,5
12	-	-	-	-	-	-	-	-	3,90	3,67	3,46
14	-	-	-	-	-	-	4,24	4	3,78	3,59	3,42
17	-	-	-	-	4,5	4,27	4,03	3,83	3,67	3,53	3,40
20	-	-	-	4,55	4,28	4,07	3,89	3,75	3,61	3,50	3,39
25	-	4,6	4,39	4,20	4,04	3,90	3,77	3,67	3,57	3,46	3,39
30	4,6	4,32	4,15	4,05	3,90	3,80	3,70	3,62	3,55	3,47	3,40
40	4,12	4,02	3,92	3,84	3,77	3,70	3,64	3,58	3,53	3,48	3,42
50	3,97	3,88	3,81	3,76	3,70	3,65	3,61	3,57	3,53	3,49	3,44
60	3,85	3,79	3,73	3,70	3,66	3,63	3,59	3,56	3,53	3,50	3,46
80	3,73	3,70	3,68	3,65	3,62	3,61	3,58	3,56	3,54	3,52	3,50
100	3,68	3,67	3,65	3,62	3,61	3,60	3,58	3,57	3,55	3,53	3,52

3. РАСЧЕТ ЧЕРВЯЧНЫХ ПЕРЕДАЧ

3.1. Выбор электродвигателя

Двигатель выбирают по методике, изложенной в п. 1.2, принимая ориентировочно КПД червячной передачи $\eta_{\text{чп}} = 0,8$.

3.2. Кинематический расчет

Задачей кинематического расчета является определение передаточного числа червячной передачи. Его находят по формуле

$$u = n_1/n_2 = \omega_1/\omega_2, \quad (3.1)$$

где n_1, ω_1 – частота вращения и угловая скорость червяка; n_2, ω_2 – частота вращения и угловая скорость червячного колеса.

Полученную величину округляют до ближайшего стандартного значения. Допускаемое отклонение от стандартного передаточного числа – 4%.

3.3. Проектировочный расчет

3.3.1. Определение допускаемых напряжений

По рекомендациям [2] ориентировочно находят скорость скольжения v_s' и выбирают материал обода червячного колеса.

Формулы для определения допускаемых контактных и изгибных напряжений приведены в табл. 3.1. Материалы группы III в редукторах общего назначения практически не используются, поэтому допускаемые напряжения для них в таблицу не внесены.

Таблица 3.1
Допускаемые напряжения для червячного колеса [2]

Группа материалов	Червяк $\leq 350HB$	Червяк $\geq 45HRC$	Нереверсивная передача	Ревверсивная передача
	[σ] _H , Н/мм ²		[σ] _F , Н/мм ²	
I	$K_{HL}C_v 0,75\sigma_B$	$K_{HL}C_v 0,9\sigma_B$	$(0,08\sigma_B + 0,25\sigma_T)K_{FL}$	$0,16\sigma_B K_{FL}$
II	$250 - 25v_s$	$300 - 25v_s$		

В формулах: K_{HL} – коэффициент долговечности при расчете по контактными напряжениям ($K_{HL} = \sqrt[8]{10^7/N_{HE}}$, где N_{HE} – эквивалентное число циклов нагружения, при $N_{HE} > 25 \cdot 10^7$ принимают $N_{HE} = 25 \cdot 10^7$); K_{FL} – коэффициент долговечности при расчете по напряжениям изгиба ($K_{FL} = \sqrt[9]{10^6/N_{FE}}$, где N_{FE} – эквивалентное число циклов нагружения, при $N_{FE} < 10^6$ принимают $N_{FE} = 10^6$, при

$N_{FE} > 25 \cdot 10^7$ принимают $N_{FE} = 25 \cdot 10^7$); C_v – коэффициент износа материала (табл. 3.2).

Таблица 3.2

Значения коэффициента износа материала [2]

v_s , м/с	1	2	3	4	5	6	7	≥ 8
C_v	1,33	1,21	1,11	1,02	0,95	0,88	0,83	0,8

3.3.2. Определение межосевого расстояния

Межосевое расстояние определяют по формуле

$$a_w = 610 \sqrt[3]{T_2 / [\sigma]_H^2}, \quad (3.2)$$

где T_2 – крутящий момент на червячном колесе, Нм.

Полученное значение a_w для редукторов общего назначения округляют до ближайшего по ряду нормальных размеров $Ra 20$.

3.3.3. Определение числа витков червяка и числа зубьев червячного колеса

Число витков червяка z_1 зависит от передаточного числа редуктора: при $8 \leq u \leq 14$ $z_1 = 4$; при $14 < u \leq 30$ $z_1 = 2$; при $u > 30$ $z_1 = 1$.

Число зубьев колеса находят по формуле

$$z_2 = z_1 u \quad (3.3)$$

с округлением до целого.

По условию отсутствия подрезания зубьев рекомендуется $z_2 \geq 26$.

3.3.4. Определение модуля зацепления и коэффициента диаметра червяка

Определяют модуль зацепления, мм,

$$m = (1,5 \dots 1,7) a_w / z_2 \quad (3.4)$$

и принимают ближайшее стандартное значение из 1-го ряда: 2,5; 3,15; 4; 5; 6,3; 8; 10; 12,5; 16. Допускается также использование 2-го ряда: 3; 3,5; 6; 7; 12.

Коэффициент диаметра определяют по рекомендации

$$q = (0,212 \dots 0,25) z_2. \quad (3.5)$$

Полученную величину округляют до стандартной из 1-го ряда: 6,3; 8; 10; 12,5; 16 или из 2-го ряда: 7,1; 9; 11,2; 14; 18. По ГОСТ 19672-74 допускается также применение значений $q = 7,5$ и $q = 12$.

3.3.5. Определение коэффициента смещения инструмента

Коэффициент смещения нарезающего инструмента находят из выражения

$$x = (a_w/m) - 0,5(q + z_2).$$

По условиям неподрезания и незаострения зубьев колеса величина смещения должна лежать в пределах $-1 \leq x \leq +1$. Если при расчете это не выполняется, то следует варьировать значениями q и z_2 . Число z_2 рекомендуется изменять в пределах 1...2 зубьев, чтобы не внести сверхнормативную погрешность в передаточное число.

3.3.6. Определение фактических значений передаточного числа и межосевого расстояния

Определяют фактическое передаточное число редуктора

$$u_{\phi} = z_2/z_1 \quad (3.6)$$

и его отклонение от заданного

$$\Delta u = (|u_{\phi} - u|/u) \cdot 100\% \leq 4\%. \quad (3.7)$$

3.3.7. Определение основных геометрических размеров передачи

Размеры червяка:

- делительный диаметр $d_1 = qm$;
- начальный диаметр $d_{w1} = m(q + 2x)$;
- диаметр вершин витков $d_{a1} = d_1 + 2m$;
- диаметр впадин витков $d_{f1} = d_1 - 2,4m$;
- делительный угол подъема витков $\gamma = \arctg(z_1/q)$;
- длина нарезаемой части червяка $b_1 = (10 + 5,5|x| + z_1)m + C$, при $x \leq 0$
 $C = 0$, при $x > 0$ $C = 100m/z_2$.

Размеры червячного колеса:

- делительный диаметр $d_2 = d_{w2} = mz_2$;
- диаметр вершин зубьев $d_{a2} = d_2 + 2m(1 + x)$;
- наибольший диаметр колеса $d_{aM2} = d_{a2} + 6m/(z_1 + 2)$;
- диаметр впадин зубьев $d_{f2} = d_2 - 2m(1,2 - x)$;
- ширина венца при $z_1 = 1, z_1 = 2$ $b_2 = 0,355a_w$; при $z_1 = 4$ $b_2 = 0,315a_w$;
- радиусы закруглений зубьев $R_a = 0,5d_1 - m$; $R_f = 0,5d_1 + 1,2m$;

– условный угол обхвата червяка венцом колеса $2\delta = \arcsin[b_2/(d_{a1}-0,5m)]$.

3.4. Проверочный расчет

3.4.1. Уточнение коэффициента полезного действия передачи

Величину КПД определяют по формуле

$$\eta_{\text{чп}} = \text{tg}\gamma / \text{tg}(\gamma + \varphi), \quad (3.8)$$

где φ - угол трения, его принимают по табл. 3.3 в зависимости от фактической скорости скольжения

$$v_s = u_{\varphi} \omega_2 d_1 / (2000 \cos\gamma). \quad (3.9)$$

В том случае, если получается $\eta_{\text{чп}} < 0,8$, следует заново подсчитать требуемую мощность электродвигателя и при необходимости назначить двигатель более мощный.

Таблица 3.3

Значения угла трения

$v_s, \text{ м/с}$	0,1	0,5	1	1,5	2	2,5	3	4	7	10	15
φ	4°30'... ...5°10'	3°10'... ...3°40'	2°30'... ...3°10'	2°20'... ...2°50'	2°00'... ...2°30'	1°40'... ...2°20'	1°30'... ...2°00'	1°20'... ...1°40'	1°00'... ...1°30'	0°55'... ...1°20'	0°50'... ...1°10'
Меньшие значения φ – для материалов группы I, большие – для материалов групп II и III.											

3.4.2. Проверка зубьев колеса по контактным напряжениям

Контактное напряжение в зацеплении сравнивают с допусковым:

$$\sigma_H = 340 \sqrt{F_{t2} K / (d_1 d_2)} \leq [\sigma]_H,$$

где $F_{t2} = 2000 T_2 / d_2$ – окружная сила на колесе, Н; K – коэффициент нагрузки (принимается в зависимости от окружной скорости колеса $v_2 = \omega_2 d_2 / 2000$, м /с: при $v_2 \leq 3$ м /с $K = 1$; при $v_2 > 3$ м /с $K = 1,1 \dots 1,3$); $[\sigma]_H$ – уточненное значение допускового контактного напряжения по фактической скорости скольжения (см. табл. 3.1, 3.2).

Допускаемая недогрузка передачи составляет 15%, допускаемая перегрузка – 5%.

3.4.3. Проверка зубьев колеса по напряжениям изгиба

Напряжение изгиба в зубе колеса сравнивают с допускаемым:

$$\sigma_F = 0,7Y_{F2}F_{t2}K/(b_2m) \leq [\sigma]_F, \quad (3.10)$$

где Y_{F2} – коэффициент формы зуба колеса, принимается по табл. 3.4 в зависимости от эквивалентного числа зубьев $z_{v2} = z_2/\cos^3\gamma$.

Таблица 3.4

Коэффициент формы зуба червячного колеса

z_{v2}	Y_{F2}	z_{v2}	Y_{F2}	z_{v2}	Y_{F2}	z_{v2}	Y_{F2}
20	1,98	30	1,76	40	1,55	80	1,34
24	1,88	32	1,71	45	1,48	100	1,30
26	1,85	35	1,64	50	1,45	150	1,27
28	1,80	37	1,61	60	1,40	300	1,24

Как правило, получается $\sigma_F \ll [\sigma]_F$, так как нагрузочная способность червячных передач ограничивается не изгибной, а контактной выносливостью.

3.4.4. Проверка редуктора на нагрев

Цель проверки (теплового расчета) – определить температуру масла в редукторе, которая не должна превышать допускаемую $[t]_M = 80 \dots 95^\circ$.

Температуру масла в корпусе червячной передачи при непрерывной работе находят по формуле

$$t_M = t_B + P_1(1 - \eta_{\text{чп}})/(K_t A), \quad (3.11)$$

где t_B – температура окружающего воздуха, принимают $t_B = 20^\circ\text{C}$; P_1 – мощность на червяке, Вт; K_t – коэффициент теплопередачи (среднее значение коэффициента $K_t = 13 \text{ Вт}/(\text{м}^2 \cdot \text{град})$); A – площадь теплоотдающей поверхности корпуса (табл. 3.5), м^2 .

Таблица 3.5

Площадь теплоотдающей поверхности корпуса червячного редуктора

Межосевое расстояние, мм	80	100	125	140	160	180	200	224
Площадь, м^2	0,19	0,24	0,36	0,43	0,56	0,67	0,8	1

При невыполнении условия $t_M \leq [t]_M$ следует увеличить площадь поверхности теплоотдачи с помощью оребрения. Если этой меры недостаточно, то необходимо предусмотреть специальные средства охлаждения (обдув корпуса вентилятором, введение в конструкцию холодильника для масла).

4. ПРОЕКТНЫЙ РАСЧЕТ ВАЛОВ

4.1. Быстроходный вал

Как правило, быстроходный вал и шестерню выполняют за одно целое - в виде вала-шестерни.

Предварительный расчет характерного диаметра производят на кручение по пониженным допускаемым напряжениям.

Диаметр входного конца вала определяют по формуле

$$d_{\text{вых1}} = \sqrt[3]{\frac{T_1}{0,2 \cdot [\tau]}} \quad (4.1)$$

где $[\tau] = 20$ МПа – допускаемое напряжение кручения.

По ГОСТ 12080-66 принимают ближайшее к вычисленному значению диаметра.

Затем назначают конструктивно диаметры ступенек под подшипники

$$d_{1п} = d_{\text{вых1}} + 5...7 \text{ мм} \quad (4.2)$$

с округлением до ближайшего стандартного диаметра внутреннего кольца подшипника.

Составляют расчётную схему вала (пример на рис. 4.1).

Определяют диаметр буртика

$$d_{1б} = d_{1п} + 5...7 \text{ мм} \quad (4.3)$$

и принимают ближайший по ряду $Ra 40$.

Предварительно выбирают подшипники. Как правило, для валов цилиндрических передач используют радиальные шариковые однорядные подшипники по ГОСТ 8338-75, а для валов конических и червячных передач – радиально-упорные роликовые по ГОСТ 333-79. Размещают подшипники в расточке корпуса редуктора, углубив их на 3... 5 мм от внутренней поверхности стенки.

Определяют усилия, действующие на вал:

усилие от соединительной муфты

$$F_{M1} = 50...125\sqrt{T_1} \quad (4.4)$$

и нагрузки, возникающие в передаче.

Находят необходимые расстояния для определения опорных реакций. Для вала, приведённого на рис. 4.1, это делается следующим образом.

Предварительно принимают по рекомендации [6] $l_1 = (1,5...2)d_{1п}$ с округлением по ряду $Ra 40$.

Вычисляют расстояние между точкой приложения усилия от муфты и реакций смежной опоры подшипника

$$L_1 = l_{M1} + l_1 - \frac{b_{п1}}{2}, \quad (4.5)$$

где $b_{п1}$ – ширина внутреннего кольца подшипника, l_{M1} – по ГОСТ 12080-66.

Определяют расстояние между серединами подшипников

$$L^B = \frac{b_{п1}}{2} + \Delta + c + b_{w1} + c + \Delta + \frac{b_{п1}}{2}, \quad (4.6)$$

где Δ – заглубление подшипника в корпус (см. выше); c – зазор между торцевой поверхностью шестерни и корпусом (рекомендуется $c \approx 0,8 \delta$, где δ – толщина стенки корпуса).

4.2. Тихоходный вал

Определяют диаметр выходного конца вала

$$d_{\text{вых1}} = \sqrt[3]{\frac{T_i}{0,2 \cdot [\tau]}}, \quad (4.7)$$

где i – номер тихоходного вала в редукторе ($i = 2$ для двухступенчатого и $i = 3$ для трёхступенчатого редуктора).

Принимаем по ГОСТ 12080-66 $d_{\text{вых1}}$ и l_{Mi} .

Конструктивно назначают диаметры шеек вала под подшипники

$$d_{\text{ш}} = d_{\text{вых}} + 7...10 \quad (4.8)$$

с округлением до ближайшего стандартного значения диаметра внутреннего кольца подшипника.

Принимают диаметр вала под зубчатым колесом

$$d_{\text{кол}} = d_{\text{ш}} + 7...10; \quad (4.9)$$

с округлением по ряду Ra 40.

Определяют усилия, действующие на вал:

Усилие от муфты

$$F_{M2} = A\sqrt{T_i}. \quad (4.10)$$

Определяем необходимые расстояния для вычисления реакций опор на тихоходном валу. Для вала, приведённого на рис. 4.2, это делается следующим

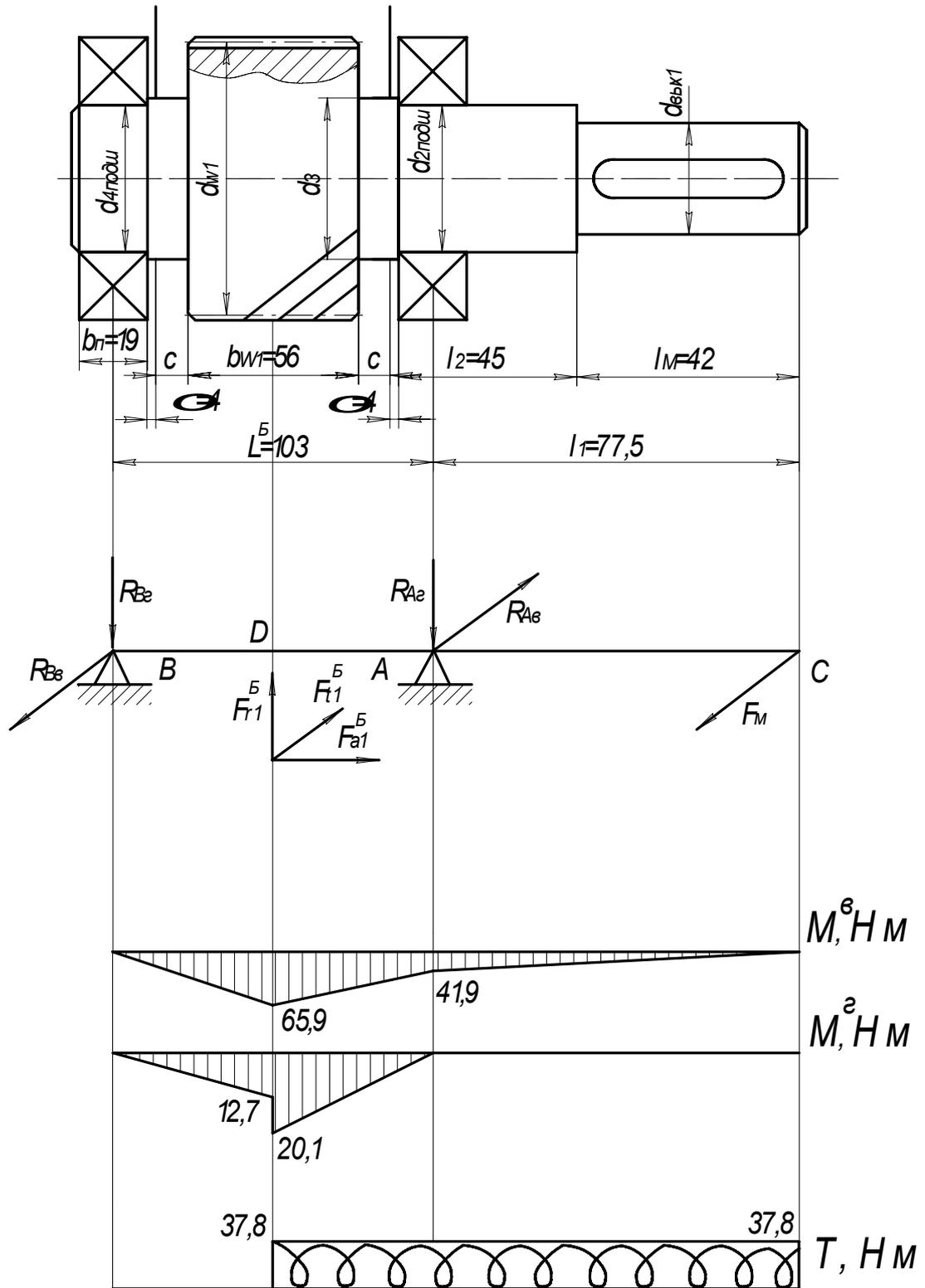


Рис. 4.1. Расчетная схема быстроходного вала

образом:

$$l_i = (0,8...1)d_{ин}; \quad (4.11)$$

$$c_i = (b_{w1} - b_{w2})/2; \quad (4.12)$$

$$L^T = b_{w2} + 2c_i + 2\Delta + b_{ин}; \quad (4.13)$$

$$L_i = l_{Mi} + l_i - c_i - b_{ин}/2. \quad (4.14)$$

4.3. Промежуточный вал

Определяют диаметр вала под колесом быстроходной ступени

$$d_{кол} = \sqrt[3]{\frac{T_2}{0,2 \cdot [\tau]}} \quad (4.15)$$

с последующим округлением по Ra 40.

Диаметр внутреннего кольца подшипника предварительно выбирают на 2...5 мм меньше, чем $d_{кол}$, с округлением до стандартного размера.

Отличительной особенностью промежуточного вала является наличие буртика между колесом быстроходной ступени и шестерней тихоходной ступени (см. рис. 4.3). Его диаметр $d_{бурт}$ принимают обычно на 3...6 мм больше, чем $d_{кол}$ по ряду Ra 40. Ширина буртика может лежать в пределах 3...5 мм.

Определение длин участков вала производят по аналогии с п.п. 4.1 и 4.2. Каких-либо дополнительных пояснений здесь не требуется.

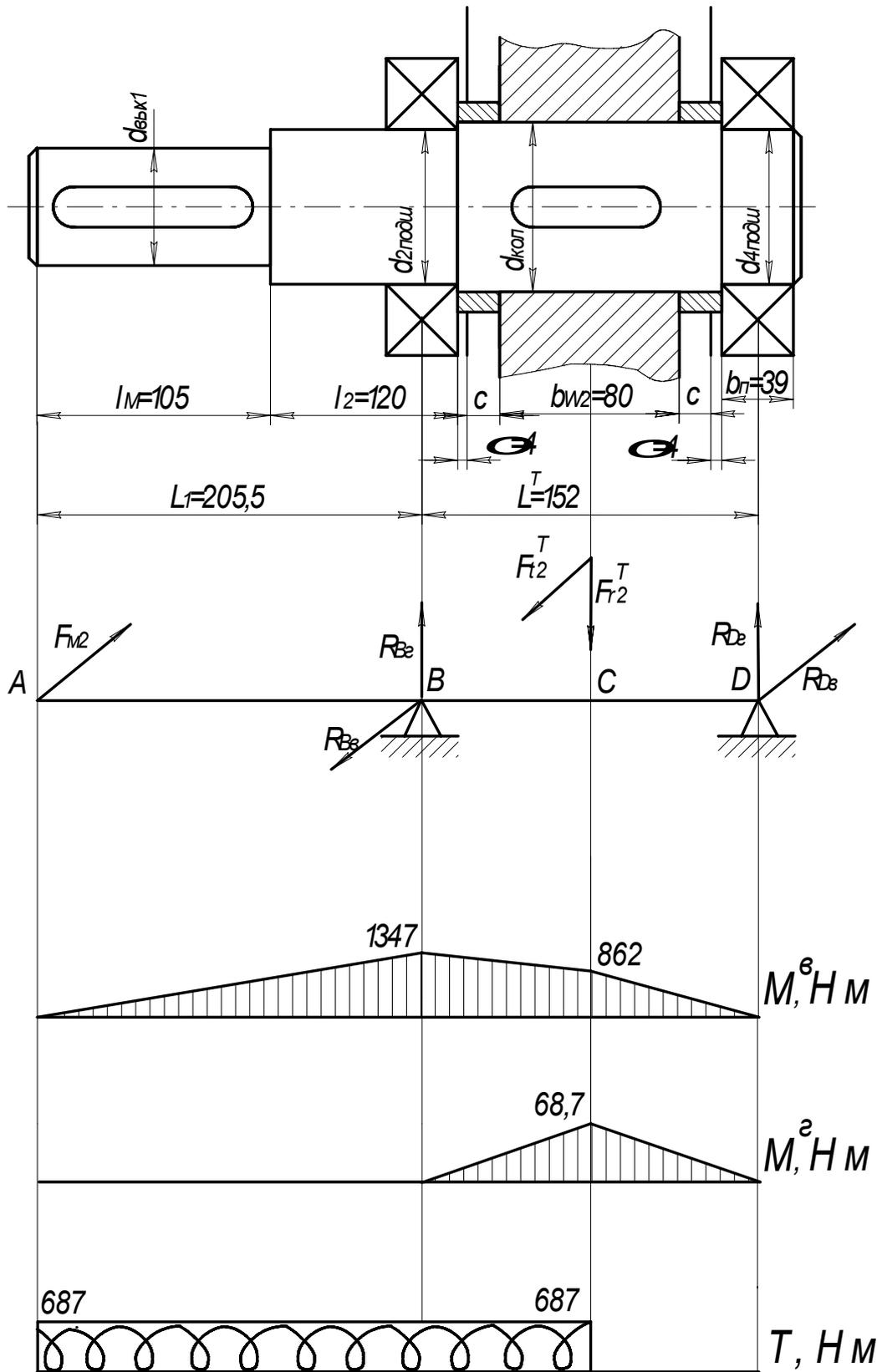


Рис. 4.2. Расчетная схема тихоходного вала

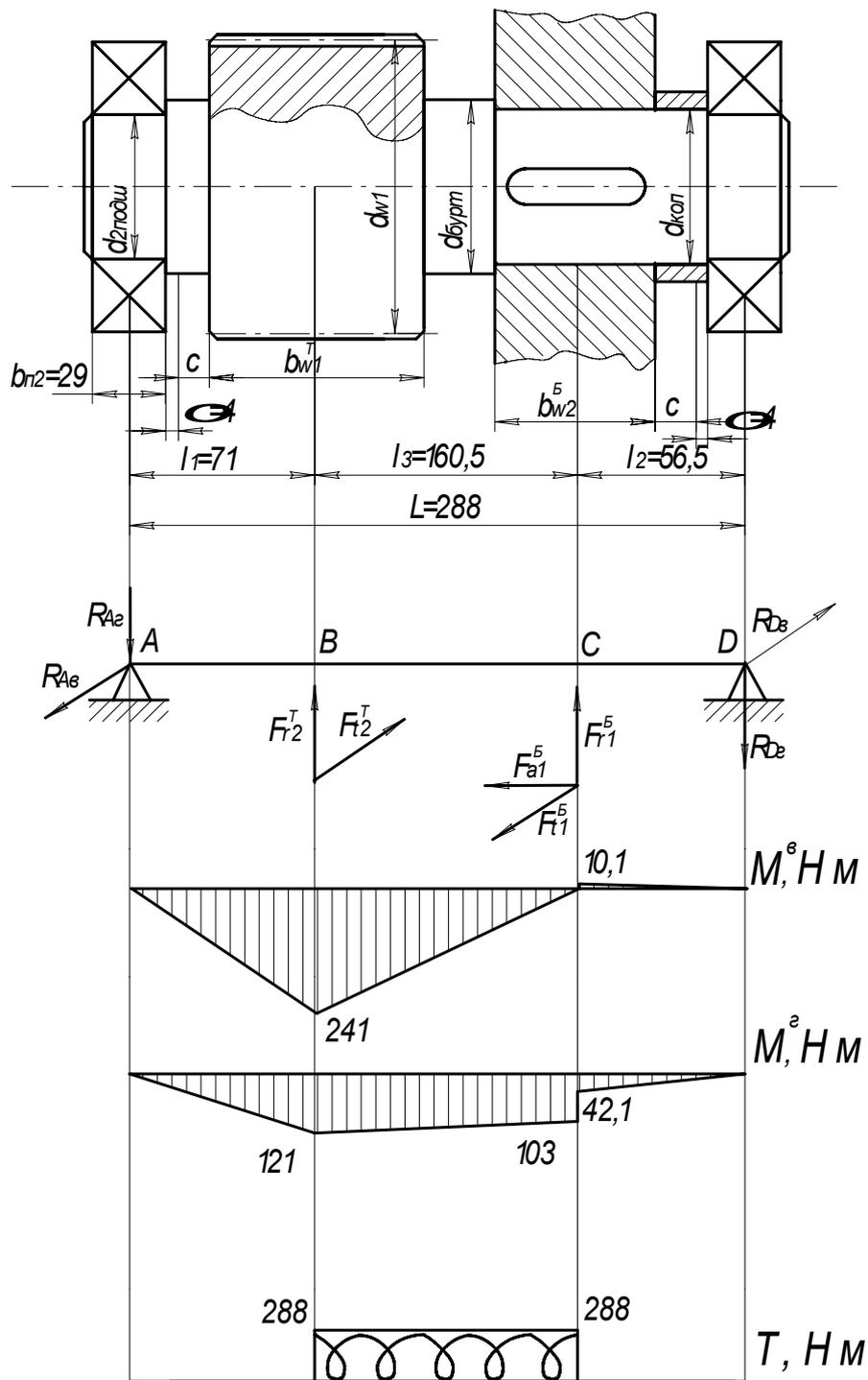


Рис. 4.3. Расчетная схема промежуточного вала

5. ПРОВЕРОЧНЫЙ РАСЧЕТ ВАЛОВ

5.1. Быстроходный вал

Проверочный расчет валов на прочность выполняют на совместное действие деформаций изгиба и кручения. При этом цель расчета – определить коэффициент запаса усталостной прочности и сравнить с допускаемым значением, равным $[S] = 1,5 \dots 2,5$ [2].

Рассмотрим расчёт на примере вала-шестерни. Материал вала тот же, что и для шестерни, так как это единая деталь (см. п.4.1).

Расчётная схема вала представляет собой балку на двух опорах с приложенными к ней сосредоточенными силами (см. рис.4.1), определёнными в п.4.1. Силы в зацеплении считают приложенными посередине ширины венца шестерни. Консольную силу от муфты в тех случаях, когда тип муфты не конкретизируется, следует считать приложенной к концу вала. Важно правильно принять направление сил. Следует помнить, что окружная сила на шестерне направлена против направления вращения шестерни, радиальная сила направлена к оси вала, направление осевой силы зависит от **направления окружной силы и направления наклона зуба**, а консольная сила направлена **параллельно** окружной и **в противоположную сторону**.

Затем определяют опорные реакции и строят эпюры изгибающих и крутящих моментов так, как это рассматривается в курсе сопротивления материалов.

Выбирают опасное сечение вала. Следует учитывать, что опасным является не то сечение, диаметр которого меньше, и не то, в котором действуют наибольшие моменты, а то, которое имеет минимальный коэффициент запаса прочности – S . Коэффициент S – комплексный показатель, учитывающий не только размеры сечения и действующие в нём нагрузки, но и свойства материала, термообработку, наличие концентраторов напряжений и пр. Поиск опасного сечения сам по себе представляет серьёзную задачу и может включать в себя расчёт нескольких «подозрительных» сечений. Поскольку курсовой проект имеет учебный характер, студентам разрешается проверять одно сечение при условии грамотного обоснования его выбора.

В качестве опасного сечения принято сечение $A-A$, в котором действуют большой изгибающий момент и имеется концентратор напряжения в виде посадки подшипника с натягом.

Вычисляют нормальное амплитудное напряжение изгиба в опасном сечении вала

$$\sigma_a = \frac{M_A \cdot 10^3}{W_x}, \quad (5.1)$$

где M_A – изгибающий момент в сечении $A-A$, Н·м; W_X – осевой момент сопротивления сечения вала, мм³ (для круглого сечения $W_X = 0,1 \cdot d_{III}^3$).

Вычисляют амплитудное касательное напряжение в опасном сечении вала

$$\tau_a = \frac{\tau_k}{2} = \frac{T_1 \cdot 10^3}{2 \cdot W_p}, \quad (5.2)$$

где T_1 – крутящий момент, Н·м; W_p – полярный момент инерции сопротивления сечения вала, мм³ (для круглого сечения $W_p = 0,2 \cdot d_{III}^3$).

Определяют коэффициент концентрации нормальных напряжений в расчетном сечении вала

$$(K_\sigma)_D = \left(\frac{K_\sigma}{K_d} + K_F - 1 \right) \frac{1}{K_y}, \quad (5.3)$$

где K_σ – коэффициент концентрации нормальных напряжений (табл. 5.1); K_d – коэффициент влияния абсолютных размеров поперечного сечения (см. табл. 5.2); K_F – коэффициент влияния шероховатости поверхности (для шероховатости поверхности по ряду Ra 2,5...0,63 мкм и $\sigma_B=700$ МПа $K_F=1,1$, при $\sigma_B=900$ МПа $K_F=1,15$); K_y – коэффициент влияния поверхностного упрочнения (для валов без поверхностного упрочнения $K_y=1$).

Определяют предел выносливости в расчетном сечении вала

$$(\sigma_{-1})_D = \frac{\sigma_{-1}}{(K_\sigma)_D}, \quad (5.4)$$

где σ_{-1} – предел выносливости гладких образцов при симметричном цикле нагружения, МПа;

$$\sigma_{-1} \approx 0,43 \cdot \sigma_B. \quad (5.5)$$

Аналогично находят коэффициент концентрации касательных напряжений

$$(K_\tau)_D = \left(\frac{K_\tau}{K_d} + K_F - 1 \right) \frac{1}{K_y}, \quad (5.6)$$

где K_τ – коэффициент концентрации касательных напряжений (табл. 5.1).

Затем определяют $(\tau_{-1})_D$ – предел выносливости в расчетном сечении при кручении

$$(\tau_{-1})_D = \frac{\tau_{-1}}{(K_\tau)_D}, \quad (5.8)$$

где $\tau_{-1} = 0,58 \cdot \sigma_{-1}$ – предел выносливости гладких образцов при кручении, МПа.

Определяют коэффициенты запаса прочности по нормальным и касательным напряжениям

$$S_\sigma = \frac{(\sigma_{-1})_D}{\sigma_a}; \quad S_\tau = \frac{(\tau_{-1})_D}{\tau_a} \quad (5.9)$$

и общий коэффициент запаса прочности в опасном сечении

$$S = \frac{S_\sigma \cdot S_\tau}{\sqrt{S_\sigma^2 + S_\tau^2}}. \quad (5.10)$$

При выполнении условия $S \geq [S]$ вал обладает требуемой усталостной прочностью. Если же условие не выполняется, то следует либо увеличить диаметр опасного сечения, либо применить сталь с более высокими прочностными характеристиками, либо устранить концентратор напряжений.

5.2. Тихоходный вал

Назначают материал вала. В отличие от вала-шестерни тихоходный вал и колесо изготавливают отдельно, поэтому вал может быть выполнен из стали другой марки, чем колесо. Следует, однако, учитывать, что для ограничения номенклатуры материалов целесообразно использовать такую же сталь.

Намечают опасное сечение. Для вала, показанного на рис. 4.2, это может быть либо *C-C*, в котором вал ослаблен шпоночным пазом, либо *B-B*, в котором действует максимальный изгибающий момент и имеется концентратор напряжений в виде посадки подшипника с натягом. Предположим для определённости, что в качестве опасного выбрано сечение *C-C*.

Вычисляют амплитудное нормальное напряжение

$$\sigma_a = \frac{M_C \cdot 10^3}{W_X}, \quad (5.11)$$

где M_C - изгибающий момент в сечении *C-C*, Н·м, равный

$$M_C = \sqrt{M_{C\beta}^2 + M_{C\gamma}^2}. \quad (5.12)$$

Моменты сопротивления W_X и W_P вычисляют по формулам, приведённым в табл. 5.3.

Таблица 5.1

Коэффициенты концентрации напряжений K_σ и K_τ

Параметры		K_σ при σ_B , Н/мм ²			K_τ при σ_B , Н/мм ²		
t/r	r/d	500	700	900	500	700	900
Для ступенчатого перехода галтелью							
≤ 1	0,01	1,35	1,4	1,45	1,3	1,3	1,3
	0,02	1,45	1,5	1,55	1,35	1,35	1,4
	0,03	1,65	1,7	1,8	1,4	1,45	1,45
	0,05	1,6	1,7	1,8	1,45	1,45	1,55
	0,1	1,45	1,55	1,65	1,4	1,4	1,45
≤ 2	0,01	1,55	1,6	1,65	1,4	1,4	1,45
	0,02	1,8	1,9	2,0	1,55	1,6	1,65
	0,03	1,8	1,95	2,05	1,55	1,6	1,65
	0,05	1,75	1,9	2,0	1,6	1,6	1,65
≤ 3	0,01	1,9	2,0	2,1	1,55	1,6	1,65
	0,02	1,95	2,1	2,2	1,6	1,7	1,75
	0,03	1,95	2,1	2,25	1,65	1,75	1,75
≤ 5	0,01	2,1	2,25	2,35	2,2	2,3	2,4
	0,02	2,15	2,3	2,45	2,1	2,15	2,25
Для шпоночных пазов, выполненных фрезой							
концевой		1,6	1,9	2,15	1,4	1,7	2,05
дисковой		1,4	1,55	1,7			
Эвольвентные шлицы		1,45	1,6	1,7	1,43	1,49	1,55
Для посадки с натягом							
Диаметр вала d , мм		K_σ/K_d			K_τ/K_d		
30		2,5	3,0	3,5	1,9	2,2	2,5
50		3,3	3,95	4,6	2,45	2,8	3,0
100		3,3	3,95	4,6	2,4	2,8	3,2
Обозначения: t – высота упорного буртика, r – радиус галтели; d – диаметр вала в опасном сечении							

Таблица 5.2

Коэффициент влияния абсолютных размеров поперечного сечения K_d

Напряженное состояние материала	Диаметр вала d , мм				
	30	40	50	70	100
Изгиб для углеродистой стали	0,88	0,85	0,81	0,76	0,71
Изгиб для легированной стали	0,77	0,73	0,70	0,67	0,62
Кручение для всех сталей					

Вычисляют амплитудное касательное напряжение

$$\tau_a = \frac{\tau_k}{2} = \frac{T_3 \cdot 10^3}{2 \cdot W_p}, \quad (5.13)$$

где T_3 - крутящий момент на валу по рис. 4.2, Н·м.

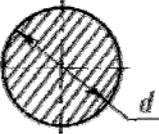
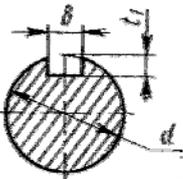
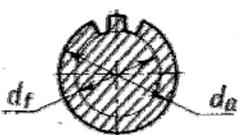
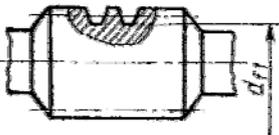
Определяют коэффициенты концентрации нормальных и касательных напряжений

$$(K_\sigma)_D = \left(\frac{K_\sigma}{K_d} + K_F - 1 \right) \frac{1}{K_y}; \quad (5.14)$$

$$(K_\tau)_D = \left(\frac{K_\tau}{K_d} + K_F - 1 \right) \frac{1}{K_y}. \quad (5.15)$$

Таблица 5.3

Осевые и полярные моменты сопротивления сечения вала

Сечение вала	$W_{\text{нетто}}$	$W_{p\text{нетто}}$
Круглое сплошное 	$0,1d^3$	$0,2d^3$
Вал со шпоночной канавкой 	$0,1d^3 - \frac{bt_1(d-t_1)^2}{2d}$	$0,2d^3 - \frac{bt_1(d-t_1)^2}{2d}$
Шлицевой вал 	$0,1 \frac{(d_a^3 + d_f^3)}{2}$	$0,2d_f^3$
Вал – червяк 	$\frac{\pi d_{f1}^3}{32}$	$\frac{\pi d_{f1}^3}{16}$

Вычисляют пределы выносливости по нормальным и касательным напряжениям в опасном сечении

$$(\sigma_{-1})_D = \frac{\sigma_{-1}}{(K_\sigma)_D}, \quad (5.16)$$

$$(\tau_{-1})_D = \frac{\tau_{-1}}{(K_\tau)_D}. \quad (5.17)$$

Затем определяют коэффициенты запаса прочности по нормальным и касательным напряжениям и общий запас усталостной прочности так, как это было показано в п. 5.1. Сравнивают значение S с допускаемой величиной и либо делают вывод о достаточной прочности вала, либо вносят в конструкцию вала необходимые изменения.

5.3. Промежуточный вал

Во многих случаях промежуточный вал выполняют заодно с шестерней тихоходной ступени (вал-шестерня), поэтому для него справедливо то, что было сказано о материале быстроходного вала в п. 6.1.

В качестве опасных могут рассматриваться сечения $B-B$ и $C-C$ (см. рис. 4.2). В сечении $B-B$ действует наибольший изгибающий момент. Но диаметр сечения $C-C$ значительно, и, кроме того, вал в этом сечении ослаблен шпоночным пазом, поэтому логично будет ожидать именно в нём возникновения максимальных напряжений.

Пусть опасным считается сечение $C-C$. Тогда ход проверки аналогичен приведённому в п. 5.2.

6. ПРОВЕРКА ПОДШИПНИКОВ

6.1. Подшипники быстроходного вала

К настоящему моменту расчёта радиальные и осевые составляющие реакций опор определены. Производят проверку подшипника в наиболее нагруженной опоре. Для вала, рассматриваемого в п.п. 4 и 5, это опора B .

Определяют отношение

$$\frac{R_a}{V \cdot R_r} = \frac{F_a}{V \cdot R_B}, \quad (6.1)$$

где R_a – осевая реакция опоры, в данном случае $R_a = F_a$.

Определяют отношение $\frac{F_a}{C_{or1}}$, где C_{or1} – табличная статистическая грузоподъёмность подшипника, и по его величине – коэффициенты e и Y . Для шариковых радиальных подшипников эти коэффициенты даны в табл. 6.1, для конических радиально-упорных подшипников – в справочных таблицах вместе с прочими параметрами.

Таблица 6.1

Коэффициенты e и Y для шариковых радиальных однорядных подшипников

R_a / C_{or}	0,014	0,028	0,056	0,084	0,11	0,17	0,28	0,42	0,56
e	0,19	0,22	0,26	0,28	0,30	0,34	0,38	0,42	0,44
Y	2,30	1,99	1,71	1,55	1,45	1,31	1,15	1,04	1,00

Смысл коэффициентов: e – коэффициент влияния осевого нагружения, Y – коэффициент осевой нагрузки.

Определяют в опоре B эквивалентную динамическую нагрузку R_{EB} .

Если $\frac{F_a}{V \cdot R_B} > e$, то эквивалентная динамическая нагрузка определяется

по формуле

$$R_{EB} = (X \cdot V \cdot R_{rB} + Y \cdot R_a) \cdot K_{\sigma} \cdot K_T, \quad (6.2)$$

где R_{rB} – радиальная нагрузка на подшипник в опоре, Н; R_a – осевая нагрузка на подшипник, Н; X – коэффициент радиальной нагрузки (для шариковых радиальных подшипников $X = 0,56$); V – коэффициент вращения ($V = 1$ – при вращающемся внутреннем кольце подшипника); K_{σ} – коэффициент безопасности (для редукторов общего назначения $K_{\sigma} = 1,3$, табл. 6.2); K_T – температурный коэффициент (при рабочей температуре подшипника до 125°C $K_T = 1$, см. табл. 6.3).

Таблица 6.2

Коэффициент безопасности K_B

Вид нагружения	K_B	Область применения
Спокойная нагрузка без толчков	1,0	Маломощные кинематические редукторы и приводы. Механизмы ручных кранов, блоков. Тали, кошки, ручные лебёдки. Приводы управления
Лёгкие толчки; кратковременные перегрузки до 125 % номинальной нагрузки	1,0-1,2	Точные зубчатые передачи. Металлорежущие станки с главным вращательным движением. Электродвигатели малой и средней мощности. Механизмы подъёма кранов. Лебёдки с механическим приводом
Умеренные толчки; вибрационная нагрузка; кратковременные перегрузки до 150 % номинальной нагрузки	1,3-1,5	Зубчатые передачи. Редукторы всех типов. Коробки передач автомобилей и тракторов. Буксы рельсового подвижного состава. Механизмы передвижения крановых тележек
То же, в условиях повышенной надёжности	1,5-1,8	Центрифуги и сепараторы. Механизмы изменения вылета стрелы кранов. Буксы и тяговые двигатели электровозов. Строгальные и долбежные станки. Мощные электрические машины
Нагрузки со значительными толчками и вибрациями; кратковременные перегрузки до 200 % номинальной нагрузки	1,8-2,5	Дробилки и копры. Кривошипно-шатунные механизмы. Валки прокатных станов. Мощные вентиляторы
Нагрузка с сильными ударами; кратковременные перегрузки до 300 % номинальной нагрузки	2,5-3,0	Тяжёлые ковочные машины. Лесопильные рамы. Рабочие роликовые конвейеры крупносортовых станов, блюмингов и слябингов

Таблица 6.3

Значение температурного коэффициента K_T

Рабочая температура подшипника, °C, до	100	125	150	175	200	225	250
K_T	1,0	1,05	1,1	1,15	1,25	1,35	1,4

Требуемую динамическую грузоподъемность определяют по формуле

$$C_{rT} = R_{EB} \sqrt[3]{573 \cdot \omega_{дв} \cdot \frac{L_h}{10^6}}; \quad (6.3)$$

где $L_h = 7 \cdot n_{см} \cdot t_{р.д.} \cdot T_{сл}$ – требуемая долговечность работы, ч.

В том случае, если $C_{rT} < C_{r1}$, где C_{r1} – табличная динамическая грузоподъемность, подшипник удовлетворяет требованию долговечности. При $C_{rT} \ll C_{r1}$ следует проверить, нельзя ли использовать подшипник более легкой серии.

Если получилось, что $\frac{F_a}{V \cdot R_B} \leq e$, то эквивалентную радиальную нагрузку

на подшипник находят по формуле для чисто радиального нагружения

$$R_{EB} = V \cdot R_B \cdot K_{\sigma} \cdot K_T. \quad (6.4)$$

Далее вычисляют требуемую динамическую грузоподъемность и сравнивают её с табличной так, как это показано выше.

6.2. Подшипники тихоходного вала

Подшипники рассматриваемого в качестве примера вала (см. п.п. 4 и 5) испытывают чисто радиальные нагрузки.

Определяют эквивалентную радиальную нагрузку на наиболее нагруженную опору B

$$R_{EB} = V \cdot R_D \cdot K_{\sigma} \cdot K_T, \quad (6.5)$$

затем – требуемую динамическую грузоподъемность

$$C_{rT} = R_{EB} \sqrt[3]{573 \cdot \omega_T \cdot \frac{L_h}{10^6}}, \quad (6.6)$$

где ω_T – угловая скорость тихоходного вала.

Найденное значение требуемой грузоподъемности сравнивают с табличной величиной (см. п. 6.1).

6.3. Подшипники промежуточного вала

Ход проверки подшипников промежуточного вала ничем не отличается от приведённых расчётов в п.п. 6.1 и 6.2 и специального рассмотрения не требует.

7. ПРОВЕРОЧНЫЙ РАСЧЕТ ШПОНОЧНЫХ СОЕДИНЕНИЙ

В редукторах общего назначения используются, как правило, ненапряжённые соединения призматической шпонкой по ГОСТ 23360-78. Полный расчёт шпоночных соединений включает в себя проверку контактирующих поверхностей на смятие и шпонки на срез. Но соотношение размеров поперечного сечения шпонки по указанному стандарту таково, что проверку шпонки на срез можно не производить. Таким образом, проверка соединения сводится к определению напряжения смятия на контактирующих поверхностях и сравнению его с допускаемой величиной.

Условие прочности записывают следующим образом:

$$\sigma_{\text{см}} = \frac{4 \cdot T \cdot 10^3}{d \cdot h \cdot l_p} \leq [\sigma_{\text{см}}] , \quad (7.1)$$

где T – крутящий момент, передаваемый соединением, Н·м; d – диаметр вала в соединении, мм; h – высота сечения шпонки, мм; l_p – рабочая длина шпонки, мм; $[\sigma_{\text{см}}]$ – допускаемое напряжение смятия (если закрепляемая деталь устанавливается на вал по посадке с натягом, то можно принимать $[\sigma_{\text{см}}] = 120$ МПа).

Из конструктивных соображений длину шпонки l принимают на 5-10 мм меньше ширины зубчатого колеса.

Рабочая длина шпонки:

со скруглёнными торцами

$$l_p = l - b ; \quad (7.2)$$

с одним скруглённым торцом

$$l_p = l - 0,5 b ; \quad (7.3)$$

с плоскими торцами

$$l_p = l , \quad (7.4)$$

где l – длина шпонки, мм; b – ширина сечения шпонки, мм.

Выбор сечения шпонки в соответствии с величиной диаметра вала в соединении, а также длины l производится по ГОСТ 23360-78.

Пример обозначения. Шпонка $b=14$ мм; $h=9$ мм; $l=45$ мм:

Шпонка $14 \times 9 \times 45$ ГОСТ 23360-78.

8. КОНСТРУИРОВАНИЕ КОРПУСНЫХ ДЕТАЛЕЙ

Корпус редуктора обычно образован двумя деталями: основанием корпуса и крышкой корпуса. Исключение составляют планетарные редукторы и червячные редукторы малых размеров, у которых корпус состоит из основной детали (собственно корпуса) и двух торцевых крышек с расточками под подшипники.

Размеры корпуса редуктора определяются числом и размерами размещённых в нём деталей, относительным их расположением, величинами зазоров между ними. Ориентировочные размеры корпуса и зазоров определяют при составлении компоновочной схемы редуктора и конструировании валов, теперь же следует их уточнить.

Корпуса современных серийных редукторов очерчивают плоскими поверхностями, все выступающие элементы вводят внутрь корпуса, проушины для транспортировки редуктора выполняют заодно с корпусом. Для удобства сборки плоскость разъёма проходит через оси валов. Плоскость разъёма и верхнюю поверхность крышки корпуса, как правило, выполняют параллельными установочной плоскости.

Для редукторов общего назначения, когда неизвестны тип и мощность двигателя, рекомендуются следующие формулы для нахождения толщины основания и крышки корпуса, а также диаметра стяжного болта:

толщину стенки основания корпуса определяют по эмпирической формуле

$$\delta = 1,34\sqrt[3]{T_3} \geq 6 \text{ мм}, \quad (8.1)$$

с округлением до целого числа;

толщину стенки крышки корпуса – по формуле

$$\delta_{кр} = 0,9 \cdot \delta \geq 6 \text{ мм}; \quad (8.2)$$

диаметр стяжного болта (винта) крепления крышки корпуса редуктора к основанию – по формуле

$$d_{ст} = 1,25\sqrt[3]{T_3} \geq 10 \text{ мм}. \quad (8.3)$$

Используют болты по ГОСТ 7798-70 или ГОСТ 7808-70.

Диаметр фундаментного болта

$$d_{\phi} = 1,25 \cdot d_{ст} \quad (8.4)$$

с округлением до ближайшего стандартного значения.

Число фундаментных болтов Z_{ϕ} принимают в зависимости от межосевого расстояния: при $a_w \leq 250$ мм $Z_{\phi}=4$, при $a_w > 250$ мм $Z_{\phi}=6$.

Диаметр установочных штифтов $d_{шт} = (0,7 \dots 0,8)d_{ст}$ с округлением до ближайшего стандартного значения по ГОСТ 3129-70 или ГОСТ 9464-70.

При размещении отверстий под болты (винты) крепления крышки к корпусу:

- отверстия располагают преимущественно по продольным сторонам;
- у бобышек стараются максимально приблизить их к отверстию под подшипник, то есть оставляют стенку минимальной толщиной, равной радиусу отверстия под стяжной болт (для увеличения жёсткости соединения);
- минимальное расстояние между стенками близко расположенных отверстий должно составлять не менее 5 мм;
- на длинных боковых сторонах помимо болтов у подшипниковых гнёзд устанавливают дополнительно болты на фланцах на расстоянии один от другого с шагом $(10 \dots 12)d$.

Диаметры отверстий под стяжные и фундаментные болты назначают по табл.8.1.

Таблица 8.1

Диаметры отверстий под стяжные и фланцевые болты

Размер	Диаметр болта							
	M10	M12	M14	M16	M18	M20	M22	M24
Диаметр отверстия под стяжной болт	11	13	15	18	20	22	24	26
Диаметр отверстия под фундаментный болт	12	14	17	19	21	24	26	29

Толщина фланца разъема корпуса и фундаментной лапы

$$h_{\phi} = d_{ст} ; \quad (8.5)$$

$$h_{л} = 1,5 \cdot d_{\phi}. \quad (8.6)$$

Ширина фланца разъема корпуса и фундаментной лапы (от наружной поверхности корпуса)

$$b_{\phi} = 2,25 \cdot d_{ст} + \delta; \quad (8.7)$$

$$b_{л} = 2,5 \cdot d_{\phi}. \quad (8.8)$$

Толщина рёбер жёсткости и проушин

$$\delta_{реб} = \delta_{кр}; \quad (8.9)$$

$$\delta_{\text{пр}} = 2,5 \cdot \delta_{\text{кр}}. \quad (8.10)$$

Размеры и конфигурацию прочих конструктивных элементов корпусных деталей редукторов (сливных отверстий, люков, и т.д.) выполняют согласно рекомендациям [2].

9. КОНСТРУИРОВАНИЕ ЗУБЧАТЫХ И ЧЕРВЯЧНЫХ КОЛЁС

9.1. Зубчатые колеса

Форма зубчатого колеса может быть плоской или со ступицей, выступающей с одной стороны. Значительно реже (в одноступенчатых редукторах) колеса делают со ступицей, выступающей в обе стороны.

При небольших диаметрах колес их изготавливают из прутка, а при больших получают заготовки свободной ковкой с последующей токарной обработкой.

Ширину ступицы $l_{СТ}$ колеса желательно принимать равной или больше ширины b_2 зубчатого венца ($l_{СТ} \geq b_2$). Принятую длину ступицы согласуют с расчетной и с диаметром посадочного отверстия d : $l_{СТ} = (0,8...1,5)d$, обычно $l_{СТ} = (1,0...1,2)d$.

Диаметр $d_{СТ}$ назначают в зависимости от материала ступицы: для стали - $d_{СТ} = (1,5...1,55)d$; для чугуна - $d_{СТ} = (1,55...1,6)d$; для легких сплавов - $d_{СТ} = (1,6...1,7)d$. Меньшие значения - для шлицевого соединения колеса с валом, большие - для шпоночного соединения и соединения с натягом.

Толщину обода колеса S принимают:

$$S = 2,2m + 0,05b_2, \quad (9.1)$$

где m - модуль зацепления, мм.

На торцах зубчатого венца (зубьях и кромках обода) выполняют фаски, величину которых находят по соотношению

$$f = (0,5...0,6)m \quad (9.2)$$

с округлением до стандартного значения.

На прямозубых зубчатых колесах фаску выполняют под углом $\alpha_{\Phi} = 45^\circ$, на косозубых колесах при твердости рабочих поверхностей менее $350HB$ - под углом $\alpha_{\Phi} = 45^\circ$, а при более высокой твердости - $\alpha_{\Phi} = 15...20^\circ$.

Острые кромки торцов ступицы также притупляют фасками, размеры которых принимают по табл. 9.1.

При серийном производстве заготовки колес получают из прутка свободной ковкой, а также ковкой в штампах. При годовом объеме выпуска колес более 50 шт. экономически оправдана ковка в простейших односторонних подкладных штампах.

Размеры фасок на торцах ступицы колеса

$d, \text{мм}$	20...30	30...40	40...50	50...80	80...120	120... ...150	150... ...250	250... ...500
$f, \text{мм}$	1,0	1,2	1,6	2,0	2,5	3,0	4,0	5,0

При готовом объеме выпуска более 100 шт. применяют двусторонние штампы. Для свободной выемки заготовок из штампа принимают значения штамповочных уклонов $\gamma \geq 7^\circ$ и радиусов закруглений $R \geq 6 \text{ мм}$.

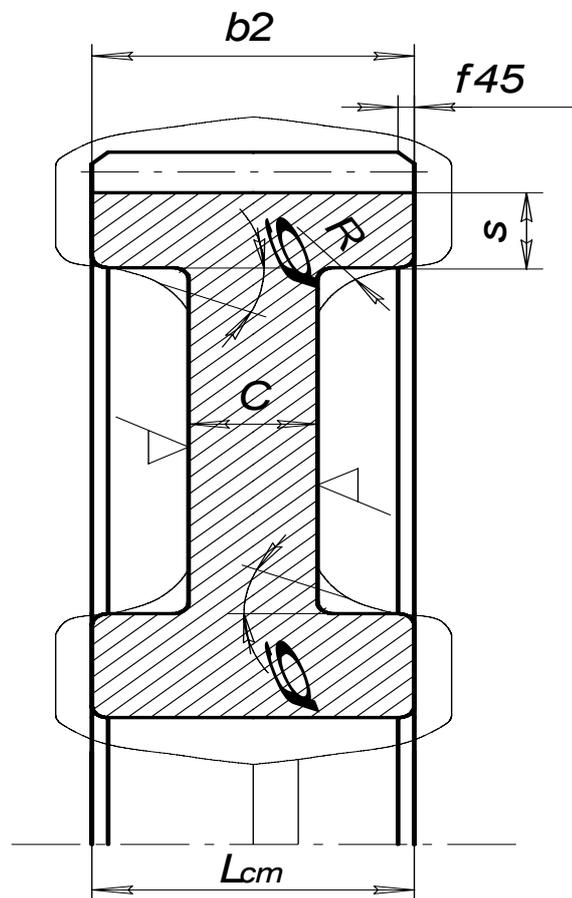


Рис. 9.1

Конструкция колеса показана на рис. 9.1.

Толщина диска

$$C \approx 0,5(S + S_{CT}) \geq 0,25b_2, \quad (9.3)$$

где $S_{CT} = 0,5(d_{CT} - d)$.

В диске обычно выполняют 4...6 технологических отверстий диаметром 18...22 мм.

Для уменьшения влияния термической обработки на точность геометрической формы зубчатые колеса при $HV > 350$ делают массивными:

$$C = (0,35...0,4)b_2. \quad (9.4)$$

Нормализованные и улучшенные колеса делают с более тонким диском ($C = 0,25b_2$).

Зубчатые колеса, вращающиеся с относительно высокой частотой ($n \geq 1000 \text{ мин}^{-1}$), обрабатывают кругом и балансируют путем высверливания отверстий на торцах обода.

Базовыми поверхностями при нарезании зубьев являются поверхность центрального отверстия и торцы зубчатого венца. Производительность возрастает при нарезании зубьев в «пакете» из двух колес и более.

9.2. Червячные колеса

Червячные колеса, как правило, выполняют составными: центр колеса (ступицу с диском) – из стали, реже из серого чугуна, а зубчатый венец (обод) – из антифрикционного материала.

В единичном и мелкосерийном производстве зубчатые венцы соединяют с центром посадкой с натягом. Для обеспечения точной сборки колеса на наружной поверхности центра предусматривают буртик, в который упирается обод при его запрессовке, и эта форма является традиционной.

Центр колеса представляет собой деталь, подобную по своей форме зубчатому колесу, но с гладкой наружной цилиндрической поверхностью, и рекомендуемые соотношения размеров ступицы и диска те же, что и в п. 9.1. Единственное отличие состоит в том, что размер S (см. рис. 9.1) определяют в зависимости от делительного диаметра червячного колеса d_2 : $S = 0,06d_2$.

Характерным размером обода является его толщина

$$S_o = 0,5(d_{f2} - d_n) = 0,05d_2, \quad (9.5)$$

где d_n – диаметр посадочной цилиндрической поверхности центра.

Пользуясь этим соотношением, находят S_o и d_n .

На зубьях обода выполняют такие же фаски, что и на зубьях цилиндрических зубчатых колес (см. п. 9.1).

10. СМАЗКА РЕДУКТОРОВ

Для снижения трения, шума и вибрации, защиты от коррозии, уменьшения интенсивности износа, отвода тепла и продуктов износа в редукторе необходимо предусмотреть надежную смазку.

Зубчатые и червячные передачи смазывают жидкими маслами картерным непроточным способом (окунанием). Сорт масла выбирают в зависимости от контактного напряжения в передаче и характерной скорости. В зубчатых передачах такой скоростью является окружная скорость шестерни (или колеса), в червячной – окружная скорость червяка. Масло для зубчатой передачи следует выбирать по табл. 10.1. Для червячных передач при $\sigma_H \leq 200$ МПа и $v \leq 5$ м/с принимают масло И-Г-С-320, а при $v > 5$ м/с – масло И-Г-С-220.

Обозначение промышленных масел состоит из четырёх знаков, каждый из которых обозначает: первый (И) – промышленное, второй – принадлежность к группе по назначению (Г – для гидравлических систем, Т – для тяжело нагруженных узлов), третий – принадлежность к группе по эксплуатационным свойствам (А – масло без присадок, С – масло с антиокислительными, антикоррозионными, противоизносными и противозадирными присадками), четвёртый (число) – класс кинематической вязкости.

Таблица 10.1

Рекомендуемые сорта масла по ГОСТ 20799-88 для зубчатых передач

Контактное напряжение, МПа	Окружная скорость, м/с		
	<2	2...5	>5
<600	И-Г-А-32	И-Г-А-32	И-Л-А-22
600...1000	И-Г-А-68	И-Г-А-46	И-Г-А-32
1000...1200	И-Г-А-68	И-Г-А-68	И-Г-А-46

Колесо цилиндрической зубчатой передачи должно окунаться в масляную ванну на глубину $h_m \approx (2 \cdot m \dots 0,25 \cdot d_2)$, где m - модуль зацепления. Колесо конической передачи – до половины зуба (до среднего диаметра зубчатого венца). В червячной передаче с нижним расположением червяка следует обеспечить погружение червяка на глубину не менее $2,2m$.

При картерном смазывании передачи и окружных скоростях более 1 м/с масло разбрызгивается зубьями колёс, а при скоростях более 3 м/с внутри корпуса образуется масляный туман. И в том, и другом случае нет необходимости предусматривать специальную смазку подшипников, т.к. они будут смазываться тем же маслом, что и передачи. Именно такие скорости характерны для большинства редукторов общего назначения.

Особое значение имеет смазка для червячных передач, т.к. без интенсивной смазки возможен перегрев редуктора. Следует обеспечить объём масла в картере V , дм^3 , равный

$$V = (0,4 \dots 0,8)P_{\text{двГ}}, \quad (10.1)$$

где $P_{\text{двГ}}$ - требуемая мощность двигателя, кВт.

Устройства для заливки масла в корпус редуктора и слива масла для замены, отдушины, маслоуказатели, а также различные типы уплотнений см. [1, 2, 4].

11. ТРЕБОВАНИЯ К РАСЧЁТНО-КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ

11.1. Расчётно-пояснительная записка

Все расчеты, схемы и эскизы, выполненные в черновике, оформляются на писчей нелинованной бумаге и комплектуются в пояснительную записку. Пояснительную записку выполняют в соответствии с общими правилами ЕСКД, соблюдение которых в курсовом проекте обязательно.

Текст записки выполняют на стандартных листах формата А4 в соответствии с ГОСТ 2.301-68. Титульный лист пояснительной записки является первым листом документа и оформляется на листе формата А4 по форме, приведенной на рис. 11.1. Высота букв в текстовых документах должна быть не менее 2,5 мм, от краев формата до границ текста рекомендуется оставлять поля: слева – 25 мм, справа – 10 мм, сверху и снизу – не менее 15 мм (ГОСТ 2.304-81).

На заглавном листе пояснительной записки (техническом задании) выполняется основная надпись по форме 2, на остальных листах – по форме 2а, а для чертежей и схем – по форме 1 с ГОСТ 2.104-68.

Пояснительная записка по курсу «Детали машин» должна содержать следующие разделы.

1. Задание на проект.
2. Содержание.
3. Выбор электродвигателя.
4. Кинематические расчеты (определение передаточных чисел и угловых скоростей валов).
5. Определение вращающих моментов на валах.
6. Выбор материалов для зубчатой пары.
7. Расчеты зубчатых, червячных или ременных передач.
8. Конструирование валов. Предварительный выбор подшипников.
9. Проверка валов на усталостную прочность.
10. Проверка подшипников по динамической грузоподъемности.
11. Проверка шпоночных соединений.
12. Выбор смазочного материала.
13. Определение основных размеров корпуса редуктора.
14. Список использованной литературы.

Каждый раздел следует начинать с новой страницы. Заголовок раздела помещать отдельной строкой прописными буквами, без переносов слов. Точку в конце заголовка не ставят. Страницы текста пронумеровать арабскими цифрами.

При выполнении текстовых материалов рекомендуется использовать глаголы во множественном числе, например: "вычисляем", "рассчитываем", "выбираем", "определяем" и т.д.

Условные буквенные обозначения механических, математических и других величин должны соответствовать стандартам.

Формулы нужно сначала записывать в буквенных выражениях, затем – в числовых значениях величин, входящих в формулу, а далее – окончательный результат вычисления. Например: "Частоту вращения барабана (рабочего органа) электролебёдки вычисляем по формуле

$$n_{p.o.} = 60 v / (\pi d) = 60 \cdot 0,65 / (3,14 \cdot 0,3) = 41,4 \text{ об/мин,}$$

где v – линейная скорость навивки каната, м/с;

d – диаметр барабана электролебёдки, м.”

Иллюстрации могут быть представлены чертежами, схемами и рисунками. На все иллюстрации должны быть ссылки в тексте. Например, "на рис. 4.1 приведены расчетная схема и эпюры моментов быстроходного вала".

Таблицы следует оформлять строго по ЕСКД.

Список использованной литературы должен состоять из библиографических ссылок на источники информации, использованные при составлении пояснительной записки. Ссылки заносятся в список в порядке упоминания в тексте цитируемого материала. Номер ссылки в тексте обозначается арабскими цифрами и заключаются в квадратные скобки. В списке должны быть указаны: фамилия и инициалы автора, название источника, город и год его издания.

Например:

1. Иванов М. Н. Детали машин. – М.: Высшая школа, 1991. 383 с.
2. Дунаев П. Ф., Леликов О. П. Конструирование узлов и деталей машин: Учеб. пособие для вузов. 5-е изд., перераб. и доп. – М.: Издательский центр «Академия», 2003. 496 с.
3. Шейнблит А. Е. Курсовое проектирование деталей машин. – М.: Высшая школа, 1991. 432 с., ил.

11.2. Рабочие чертежи

11.2.1. Сборочный чертеж редуктора

Разработка сборочного чертежа редуктора производится в соответствии с ГОСТ 2.109-73. Сборочный чертёж выполняется на чертёжной бумаге формата А1 карандашом в стандартном масштабе, например, 1:1; 1:2; 1:2,5; 1:4 таким образом, чтобы было использовано всё поле чертежа. Чертеж должен содержать три проекции редуктора (главный вид, разрез по плоскости разъёма и вид сбоку), а при необходимости – дополнительные виды и разрезы; номера позиций для спецификации, размеры; текстовую часть и основную надпись. Он должен

давать полное представление о конструкции. Допускается компьютерное исполнение чертёжно-технической документации.

Размеры на сборочном чертеже наносят в соответствии с ГОСТ 2.307-68. Линейные размеры и отклонения линейных размеров на чертеже указывают в миллиметрах без обозначения единиц величин.

Сборочный чертёж должен содержать следующие размеры:

– габаритные;

– присоединительные (диаметр и количество отверстий для крепления редуктора к раме или фундаменту, а также размеры между осями этих отверстий; расстояния от установочной плоскости до осей быстроходного и тихоходного валов; межосевые и конусные расстояния; диаметры и длины выходных концов валов);

– посадочные (колёс и подшипников на валы, подшипники в корпус и т.п.).

Кроме того, могут быть указаны и другие размеры, позволяющие, по мнению разработчика, пояснить особенности работы механизма, форму деталей, способы их закрепления и т.д.

Номера позиций (ГОСТ 2.109-79) указывают на полках линий-выносок, проводимых от элементов сборочной единицы на сборочном чертеже. Номера позиций располагают параллельно основной надписи чертежа вне контура изображения и группируют в колонки или строки. Размер шрифта номеров позиций должен быть на один – два размера больше, чем для размерных чисел. Допускается делать общую линию-выноску с вертикальным расположением номеров позиций (например, для группы крепёжных деталей). Линии-выноски не должны пересекаться, не должны пересекать размерные линии. Началом линии-выноски является точка, отмечающая тот элемент, от которого линия-выноска проводится.

Текстовую часть располагают на свободном поле чертежа над основной надписью в колонки шириной 180...185 мм. Текстовая часть сборочного чертежа редуктора содержит обязательные и дополнительные данные. Обязательные данные - техническая характеристика и технические требования, дополнительные данные - характеристика зацепления.

Техническая характеристика содержит: фактическое передаточное число редуктора; вращающий момент на тихоходном валу; частоту вращения быстроходного вала.

Технические требования включают: указания размеров для справок, обозначение смазочных материалов для передач и подшипников, требования к покрытию плоскости разъёма основания и крышки корпуса редуктора.

Характеристика зацепления (основные геометрические параметры передач) оформляются в виде таблицы, которая заполняется для зубчатых (червячных) передач следующим образом.

Числа зубьев шестерни (червяка) и колеса.

Модуль зацепления.

Угол наклона линии зубьев (витков червяка).

Ширина шестерни (длина нарезаемой части червяка) и колеса.
Степень точности передачи.

Примеры.

Характеристика зацепления

Модуль		m	1,5
Число зубьев	шестерни	z_1	20
	колеса	z_2	80
Угол наклона зубьев		β	0°
Ширина	шестерни	b_1	50
	колеса	b_2	45
Степень точности		-	8-B

Техническая характеристика

1. Передаточное число редуктора $u = 4$.
2. Вращающий момент на тихоходном валу $T_2 = 516 \text{ Н} \cdot \text{м}$.
3. Частота вращения быстроходного вала $n_1 = 790 \text{ об/мин}$.

Технические требования

1. Размеры для справок.
2. В редуктор залить масло И-Г-А-46 ГОСТ 20799-88 до верхней отметки маслоуказателя.
3. Допускается эксплуатировать редуктор с отклонением от горизонтального положения на угол до 5° . При этом должен быть обеспечен уровень масла, достаточный для смазки зацепления.

11.2.2. Чертежи деталей

Чертежи деталей редуктора должны иметь все данные, необходимые для изготовления деталей в соответствии с их назначением и расчётами, т.е. содержать указания по форме, размерам, точности, шероховатости поверхностей, материалу и термообработке, отделке и пр.

Детали изображают в положении, в котором деталь устанавливают на станке. Например, ось детали вращения (вал, зубчатое колесо, шкив) располагают параллельно основной надписи чертежа. Изображение детали на чертеже должно иметь минимальное число проекций, видов, разрезов и сечений. Каждый размер следует указывать на чертеже лишь один раз.

Размеры на чертежах не допускается наносить в виде замкнутой цепи, за исключением случаев, когда один из размеров указан как справочный. Справочные размеры на чертеже обозначаются знаком «*». Размеры, относящиеся к одному конструктивному элементу, следует группировать в одном месте. Час-

той ошибкой является включение в общую размерную цепь ширины фасок и канавок. Размеры фасок и канавок должны быть указаны отдельно. Размерные числа на нескольких параллельных или концентрических размерных линиях следует наносить в шахматном порядке. Размеры элементов деталей, обрабатываемых совместно, заключают в квадратные скобки. В технических требованиях в этом случае записывают: «Обработку по размерам в квадратных скобках производить совместно с деталью №...».

В редукторах общего назначения применяют преимущественно **посадки** в системе отверстия с основным отверстием H . Для посадок с зазором рекомендуют применять неосновные валы с допусками по f, g, h ; для переходных посадок – j_s, k, m, n ; для посадок с натягом – p, r, s . Это следует учитывать при постановке на чертеже размеров детали.

Важным качеством детали является шероховатость её поверхностей. Основные параметры **шероховатости** по ГОСТ 2789-73:

Ra – среднее арифметическое отклонение профиля, мкм;

Rz – высота неровностей профиля, мкм;

t_p – относительная опорная длина профиля.

Параметр Ra является основным для деталей в машиностроении. Параметр Rz назначают для несопрягаемых, грубо обработанных поверхностей, а также для поверхностей, получаемых литьём, ковкой, чеканкой. Значение шероховатости указывают в соответствии с требованиями ГОСТ 2.309-73.

11.3. Спецификация

Спецификацию выполняют на отдельных листах формата А4 по ГОСТ 2.108-68. Первый лист спецификации содержит основную надпись по форме 2, последующие – по форме 2а. В общем случае спецификация состоит из следующих разделов: документация, сборочные единицы, детали, стандартные изделия, материалы.

В графе «Поз.» (позиция) указывают порядковый номер составных частей редуктора.

В графе «Обозначение» – шифр чертежа составной части (для стандартных изделий эту графу не заполняют).

В графе «Наименование» – наименование изделия (для стандартных изделий указывают наименование и условное обозначение по соответствующему стандарту).

В графе «Кол.» – количество составных частей в редукторе.

Сведения о документации, сборочных единицах, деталях и стандартных изделиях оформляются в виде разделов с заголовками. Заголовки пишут в графе «Наименование» и подчёркивают сплошной линией. Выше и ниже заголовка оставляют по одной свободной строке. В разделе «Документация» указывают наименование документа, например, «Сборочный чертёж» и «Пояснительная записка». В разделе «Сборочные единицы» и «Детали» запись производят в по-

рядке возрастания обозначений, например, «Указатель жезловый», «Крышка-отдушина», «Корпус», «Крышка корпуса». В разделе «Стандартные изделия» приводят наименования и обозначения изделий в соответствии со стандартом, по группам изделий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шейнблит А.Е. Курсовое проектирование деталей машин. – М.: Высш. шк., 1991.- 432 с.
2. Дунаев П.Ф., Леликов О.П. Детали машин. Курсовое проектирование. – М.: Машиностроение., 2004. – 560 с.
3. Проектирование механических передач./С.А. Чернавский, Г.А. Снесарев, Б.С. Козинцов и др. – М.: Машиностроение, 1984.- 560 с.
4. Анурьев В.И. Справочник конструктора-машиностроителя. В 3-х т. – М.: Машиностроение, 1980.

ПРИЛОЖЕНИЕ 1

Таблица П.1.1

Номинальные линейные размеры (ГОСТ 6636-69)

R _a 5	R _a 10	R _a 20		R _a 40				Дополнительные размеры			
				1,0	1,05	1,1	1,15				
1,0	1,0	1,0	1,1	1,0	1,05	1,1	1,15	1,25	1,35	1,45	1,55
	1,2	1,2	1,4	1,2	1,3	1,4	1,5				
1,6	1,6	1,6	1,8	1,6	1,7	1,8	1,9	1,05	1,75	1,85	1,95
	2,0	2,0	2,2	2,0	2,1	2,2	2,4	2,05	2,15	2,3	
2,5	2,5	2,5	2,8	2,5	2,6	2,8	3,0		2,7	2,9	3,1
	3,2	3,2	3,6	3,2	3,4	3,6	3,8	3,3	3,5	3,7	3,9
4,0	4,0	4,0	4,5	4,0	4,2	4,5	4,8	4,1	4,4	4,6	4,9
	5,0	5,0	5,6	5,0	5,3	5,6	6,0	5,2	5,5	5,8	6,2
6,3	6,3	6,3	7,1	6,3	6,7	7,1	7,5	6,5	7,0	7,3	7,8
	8,0	8,0	9,0	8,0	8,5	9,0	9,5	8,2	8,8	9,2	9,8
10	10	10	11	10	10,5	11	11,5	10,2	10,8	11,2	11,8
	12	12	14	12	13	14	15	12,5	13,5	14,5	15,5
16	16	16	18	16	17	18	19	16,5	17,5	18,5	19,5
	20	20	22	20	21	22	24	20,5	21,5	23	
25	25	25	28	25	26	28	30		27	29	31
	32	32	36	32	34	36	38	33	35	37	39
40	40	40	45	40	42	45	48	41	44	46	49
	50	50	56	50	53	56	60	52	55	58	62
63	63	63	71	63	67	71	75	65	70	73	78
	80	80	90	80	85	90	95	82	88	92	98
100	100	100	110	100	105	110	120	102	108	112	115
	125	125	140	125	130	140	150	118	135	145	155
160	160	160	180	160	170	180	190	165	175	185	
	200	200	220	200	210	220	240	205	215	230	
250	250	250	280	250	260	280	300	270	290	300	315
	320	320	360	320	340	360	380	330	350	370	390
400	400	400	450	400	420	450	460	410	440	460	490
	500	500	560	500	500	560	600	515	545	580	615

Примечание. Настоящий стандарт устанавливает ряды нормальных линейных размеров (диаметров, длин, высот и др.) в интервале 0,001-20000мм; при выборе размеров предпочтение должно отдаваться ряду с более крупной градацией (ряд R_a5 следует предпочесть ряду R_a10, ряд R_a10 – ряду R_a20 и т.д.).

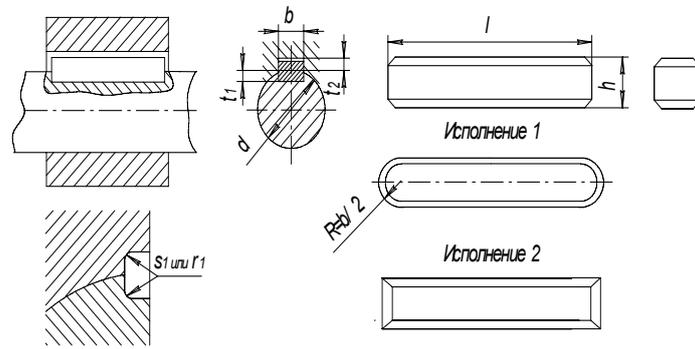
ПРИЛОЖЕНИЕ 2

Числовые значения параметра шероховатости R_a

Вид поверхности	R_a , мкм
Поверхности валов для соединений с натягом	0,8
Поверхности валов под резиновые манжеты	0,4
Поверхности шпоночных пазов на валах:	
рабочие	3,2
нерабочие	6,3
Посадочные поверхности валов и корпусов под подшипники качения класса точности 0 при:	
d или D до 50 мм	0,8
d или D св. 50 мм	1,6
Торцы заплечиков валов и корпусов для базирования подшипников качения класса точности 0	1,6
Рабочие поверхности зубьев зубчатых колёс внешнего зацепления для степени точности:	
6	0,4
7	0,8
8	1,6
9	3,2
Торцы заплечиков валов для базирования зубчатых, червячных колёс, при отношении длины отверстия ступицы к его диаметру:	
$l/d < 0,7$	1,6
$l/d \geq 0,7$	3,2
Поверхности отверстий ступиц для соединений с натягом	1,6
Торцы ступиц зубчатых, червячных колёс, базирующихся по торцу заплечиков валов, при отношении длины отверстия в ступице к его диаметру:	
$l/d < 0,7$	1,6
$l/d \geq 0,7$	3,2
Торцы ступиц зубчатых, червячных колёс, по которым базируются подшипники качения класса точности 0	1,6
Поверхности шпоночных пазов в отверстиях колёс, шкивов	
рабочие	1,6
нерабочие	3,2
Свободные (нерабочие) торцовые поверхности зубчатых, червячных колёс, поверхности выступов зубьев колёс, витков червяков, опорные поверхности под головки болтов, винтов, гаек	6,3
Поверхности отверстий под болты, винты, шпильки	12,5

ПРИЛОЖЕНИЕ 3

Призматическая шпонка (ГОСТ23360-78)



<i>d</i>	<i>b</i>	<i>h</i>	<i>t</i> ₁	<i>t</i> ₂	<i>l</i>		<i>s</i> ₁ или <i>r</i> ₁	<i>s</i> или <i>r</i>		
					от	до				
Св. 12 до 17	5	5	3	2,3	-	-	-	-		
» 17 » 22	6	6	3,5	2,8	-	-	-	-		
» 22 » 30	8	7	4,0	3,3	-	-	-	-		
От 30 до 38	10	9	5,5	3,8	22	110	0,25...0,4	0,4...0,6		
Св. 38 » 44	12	11	7	4,4	28	140				
» 44 » 50	14	12	7,5	4,9	38	160				
» 50 » 58	16	14	9	5,4	45	180				
» 58 » 65	18	16	10	6,5	50	200				
» 65 » 75	20	18	11	7,4	56	220	0,4... 0,6	0,6...0,8		
» 75 » 85	22	20	12	8,4	63	250				
» 85 » 95	25	22	13	9,4	70	280				
» 95 » 110	28	25	15	10,4	80	320				
» 110 » 130	32	28	17	11,4	90	360				
» 130 » 150	36	32	20	12,4	100	400	0,7... 1,0	1,0...1,2		
» 150 » 170	40	36	22	14,4						
» 170 » 200	45	40	25	15,4					110	450
» 200 » 230	50	45	28	17,4					125	500
» 230 » 260	56	50	31	19,5	140	500	1,2... 1,6	1,6...2,0		
» 260 » 290	63	60	36	24,5	160					
» 290 » 330	70	65	39	26,5	180					
» 330 » 380	80	75	44	31,5	200		2,0... 2,5	2,5...3,0		
» 380 » 440	90	85	49	36,5	220					
» 440 » 500	100	95	54	41,5	250					

Примечания: 1. Допускается в отдельных обоснованных случаях применять меньшие размеры сечений шпонок на валах больших диаметров, за исключением выходных концов валов.

2. Поля допусков ширины и высоты шпонки принимаются соответственно *h9* и *h11*.

3. Материал шпонок по ГОСТ 10748 – 79 сталь с временным сопротивлением разрыву не менее 590 МПа : материал шпонок по ГОСТ 24071 – 80 сталь чистотянутая для сегментных шпонок по ГОСТ 8786 – 68: допускается также применение вышеуказанной стали.

4. Стандартные длины шпонок: 6; 8; 10; 12; 14; 16; 20; 22; 25; 28; 32; 36; 40; 45; 50; 56; 63; 77; 80; 90; 100; 110; 125; 140; 160; 180; 200; 220; 250мм.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1. РАСЧЕТ ЗАКРЫТЫХ ЦИЛИНДРИЧЕСКИХ ПЕРЕДАЧ.....	4
1.1. Общие положения.....	4
1.2. Выбор электродвигателя.....	5
1.3. Допускаемые напряжения.....	7
1.3.1. Допускаемые контактные напряжения при расчете на выносливость.....	7
1.3.2. Допускаемые напряжения изгиба при расчете на выносливость.....	10
1.4. Распределение передаточного числа.....	11
1.5. Проектный расчет на контактную выносливость.....	14
1.5.1. Межосевое расстояние.....	14
1.5.2. Выбор модуля и числа зубьев.....	17
1.5.3. Определение ширины колеса.....	19
1.6. Проверочный расчет на выносливость по контактным напряжениям.....	20
1.7. Проверочный расчет на выносливость по напряжениям изгиба.....	21
1.8. Проверочный расчет прочности зубьев при перегрузках.....	22
1.9. Особенности расчета открытых цилиндрических зубчатых передач.....	26
2. РАСЧЕТ ЗАКРЫТОЙ КОНИЧЕСКОЙ ЗУБЧАТОЙ ПЕРЕДАЧИ.....	27
2.1. Проектный расчет.....	27
2.1.1. Внешний делительный диаметр колеса.....	27
2.1.2. Внешнее конусное расстояние и углы делительных конусов.....	27
2.1.3. Ширина колес и модуль.....	28
2.1.4. Числа зубьев колес.....	28
2.1.5. Выбор коэффициентов смещения.....	29
2.2. Проверочный расчет.....	30
2.2.1. Проверочный расчет по контактным напряжениям.....	30
2.2.2. Проверочный расчет на выносливость по напряжениям изгиба.....	31
3. РАСЧЕТ ЧЕРВЯЧНЫХ ПЕРЕДАЧ.....	33
3.1. Выбор электродвигателя.....	33
3.2. Кинематический расчет.....	33
3.3. Проектировочный расчет.....	33
3.3.1. Определение допускаемых напряжений.....	33
3.3.2. Определение межосевого расстояния.....	34
3.3.3. Определение числа витков червяка и числа зубьев червячного колеса.....	34
3.3.4. Определение модуля зацепления и коэффициента диаметра червяка.....	34
3.3.5. Определение коэффициента смещения инструмента.....	35
3.3.6. Определение фактических значений передаточного числа и межосевого расстояния.....	35
3.3.7. Определение основных геометрических размеров передачи.....	35

3.4. Проверочный расчет.....	36
3.4.1. Уточнение коэффициента полезного действия передачи.....	36
3.4.2. Проверка зубьев колеса по контактным напряжениям.....	36
3.4.3. Проверка зубьев колеса по напряжениям изгиба.....	37
3.4.4. Проверка редуктора на нагрев.....	37
4. ПРОЕКТНЫЙ РАСЧЕТ ВАЛОВ.....	38
4.1. Быстроходный вал.....	38
4.2. Тихоходный вал.....	39
4.3. Промежуточный вал.....	41
5. ПРОВЕРОЧНЫЙ РАСЧЕТ ВАЛОВ.....	44
5.1. Быстроходный вал.....	44
5.2. Тихоходный вал.....	46
5.3. Промежуточный вал.....	49
6. ПРОВЕРКА ПОДШИПНИКОВ.....	50
6.1. Подшипники быстроходного вала.....	50
6.2. Подшипники тихоходного вала.....	52
6.3. Подшипники промежуточного вала.....	53
7. ПРОВЕРОЧНЫЙ РАСЧЕТ ШПОНОЧНЫХ СОЕДИНЕНИЙ.....	54
8. КОНСТРУИРОВАНИЕ КОРПУСНЫХ ДЕТАЛЕЙ.....	55
9. КОНСТРУИРОВАНИЕ ЗУБЧАТЫХ И ЧЕРВЯЧНЫХ КОЛЕС.....	58
9.1. Зубчатые колёса.....	58
9.2. Червячные колёса.....	60
10. СМАЗКА РЕДУКТОРОВ.....	61
11. ТРЕБОВАНИЯ К РАСЧЕТНО-КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ.....	63
11.1. Расчетно-пояснительная записка.....	63
11.2. Рабочие чертежи.....	65
11.2.1. Сборочный чертеж редуктора.....	65
11.2.2. Чертежи деталей.....	67
11.3. Спецификация.....	68
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	70
ПРИЛОЖЕНИЕ 1.....	71
ПРИЛОЖЕНИЕ 2.....	72
ПРИЛОЖЕНИЕ 3.....	73

Министерство науки и высшего образования РФ
ФГБОУ ВО
«Уральский государственный горный университет»

В. М. Таугер

**ДЕТАЛИ
МЕХАТРОННЫХ МОДУЛЕЙ И РОБОТОВ**

Учебное пособие

Екатеринбург
2021

УДК 681.323-181.4(075)

Рецензент:

Таугер В. М.

Детали мехатронных модулей и роботов: учебное пособие. – Екатеринбург: УГГУ, 2021. – 107 с.: ил.

Изложены методы расчета деталей и преобразователей движения, необходимые для конструирования мехатронных модулей и роботов; рассмотрены конструктивные отличия основных деталей преобразователей движения мехатронных модулей от деталей передач общего назначения.

Учебное пособие предназначено для студентов-бакалавров направления «Мехатроника и робототехника».

Рекомендовано редакционно-издательским советом университета.

ОГЛАВЛЕНИЕ

Введение.....	5
1. Соединения.....	6
1.1. Основные понятия.....	6
1.2. Шпоночные соединения.....	6
1.3. Зубчатые (шлицевые) соединения.....	8
1.4. Резьбовые соединения.....	10
1.4.1. Классификация резьб.....	10
1.4.2. Геометрические параметры треугольной резьбы.....	11
1.4.3. Предотвращение самоотвинчивания в резьбовых соединениях.....	11
1.4.4. Расчет резьбовых соединений на прочность.....	12
1.4.5. Материалы и допускаемые напряжения.....	18
1.5. Заклепочные соединения.....	19
1.5.1. Разновидности заклепочных соединений.....	19
1.5.2. Расчет заклепочного соединения на прочность.....	20
1.6. Сварные соединения.....	22
1.6.1. Виды сварки.....	22
1.6.2. Соединения ручной электродуговой сваркой.....	23
1.6.3. Расчет сварных соединений на прочность.....	25
1.7. Соединения с натягом.....	28
2. Передачи.....	29
2.1. Общие сведения.....	29
2.2. Ременные передачи.....	30
2.3. Цилиндрические зубчатые передачи.....	32
2.3.1. Геометрические параметры цилиндрических зубчатых передач.....	32
2.3.2. Силы, действующие в зубчатом зацеплении.....	37
2.3.3. Виды разрушения зубьев.....	38
2.3.4. Материалы зубчатых передач.....	39
2.3.5. Методика расчета закрытой зубчатой передачи.....	40
2.4. Конические зубчатые передачи.....	56
2.4.1. Геометрические параметры конических зубчатых передач.....	56
2.4.2. Силы в конической зубчатой передаче.....	59
2.4.3. Методика расчета конической зубчатой передачи.....	60
2.5. Червячные передачи.....	65
2.5.1. Геометрические параметры червячных передач.....	65
2.5.2. Силы в червячной передаче.....	68
2.5.3. Материалы червячных передач.....	69
2.5.4. Методика расчета червячной передачи.....	69
2.6. Планетарные зубчатые передачи.....	74
2.6.1. Общие сведения о планетарных передачах.....	74
2.6.2. Передаточное число и условия существования планетарного механизма.....	77
2.6.3. Материалы планетарных передач.....	78

2.6.4. Методика расчета планетарной передачи.....	78
2.7. Волновые зубчатые передачи.....	81
2.7.1. Общие сведения.....	81
2.7.2. Передаточное число волновой передачи.....	83
2.7.3. Геометрические параметры волновой передачи.....	83
2.7.4. Материалы волновых передач.....	85
2.7.5. Методика расчета волновой передачи.....	85
3. Подшипники.....	93
3.1. Классификация подшипников по виду трения.....	93
3.2. Конструкции и классификация подшипников качения.....	94
3.3. Методика расчета подшипников качения.....	95
3.3.1. Исходные данные для расчета.....	95
3.3.2. Предварительный выбор подшипников.....	95
3.3.3. Проверочный расчет подшипников по динамической грузоподъемности.....	96
4. Валы и оси.....	104
4.1. Общие сведения.....	104
4.2. Методика расчета валов.....	104
4.2.1. Исходные данные.....	104
4.2.2. Проектировочный расчет.....	104
4.2.3. Разработка расчетной схемы.....	105
4.2.4. Построение эпюр изгибающих и крутящих моментов в сечениях вала.....	106
4.2.5. Проверка вала на усталостную прочность.....	106
Библиографический список.....	107

ВВЕДЕНИЕ

Конструирование мехатронных модулей и роботов невозможно без знания методов расчета и конструирования их основных деталей и передач, называемых также преобразователями движения. В данном учебном пособии рассматриваются механические преобразователи движения, в качестве которых используют зубчатые (в том числе планетарные), червячные, волновые передачи.

В процессе разработки передачи конструктор должен учитывать ряд особых требований, предъявляемых к мехатронным модулям и роботам и решающим образом влияющих на выбор типа, расчет и конструкцию передачи.

В число требований часто входит ограничение кинематической погрешности и мертвого хода рабочего органа, способы вычисления которых также приведены в учебном пособии.

Для надежного усвоения студентом материала пособия необходимы достаточно твердые и обширные знания дисциплин «Теоретическая механика» и «Сопротивление материалов».

1. СОЕДИНЕНИЯ

1.1. Основные понятия

Соединением называется неподвижное сопряжение деталей между собой.

По принципу **неразрушаемости при разборке** различают соединения разъемные и неразъемные. К **разъемным** относят соединения, допускающие разборку без разрушения или повреждения элементов, а к **неразъемным** – соединения, разборка которых невозможна без разрушения или повреждения элементов. В число **элементов** соединения включают как сами соединяемые детали, так и изделия или материалы, которыми данное соединение обеспечивается.

Из разъемных соединений наибольшее распространение получили следующие:

- шпоночные;
- зубчатые (шлицевые);
- резьбовые;
- соединения с некруглым валом.

Из неразъемных соединений наиболее часто используются:

- заклепочные;
- сварные;
- соединения с натягом (фрикционные);
- клеевые.

Соединения с некруглым валом и клеевые в настоящем учебном пособии рассматриваться не будут. Для их расчета следует обратиться к учебно-методической и технической литературе [1, 2].

1.2. Шпоночные соединения

Шпоночные соединения образуются с помощью специальной крепежной детали, которая называется **шпонкой**. Шпонка располагается между соединяемыми деталями и передает нагрузку (усилие или крутящий момент) с одной детали на другую.

Различают **напряженные** и **ненапряженные** шпоночные соединения. В первых напряжения в шпонке и соединяемых деталях возникают уже в процессе сборки, во вторых же в отсутствие полезной нагрузки на соединение напряжения в шпонке и деталях практически равны нулю. В мехатронных модулях (ММ) и роботах, как правило, применяются ненапряженные соединения.

Известны ненапряженные соединения круглой шпонкой, сегментной шпонкой и призматической шпонкой.

Круглая шпонка представляет собой цилиндрическую деталь (штифт), входящую по переходной посадке в отверстия соединяемых деталей. В

большинстве случаев соединение служит для обеспечения точности сборки и не предназначено для передачи рабочей нагрузки.

Боковая поверхность **сегментной шпонки**, как показывает само название, представляет собой сегмент окружности. Соединение технологично в изготовлении и способно передавать небольшие нагрузки. Его расчет приведен в [1].

Наибольшие нагрузки могут быть реализованы в соединении **призматической шпонкой**. Используются шпонки со скругленными торцами, с одним скругленным торцом и с плоскими торцами (рис. 1.1, а). Размеры b , h и l оговорены ГОСТ 23360.

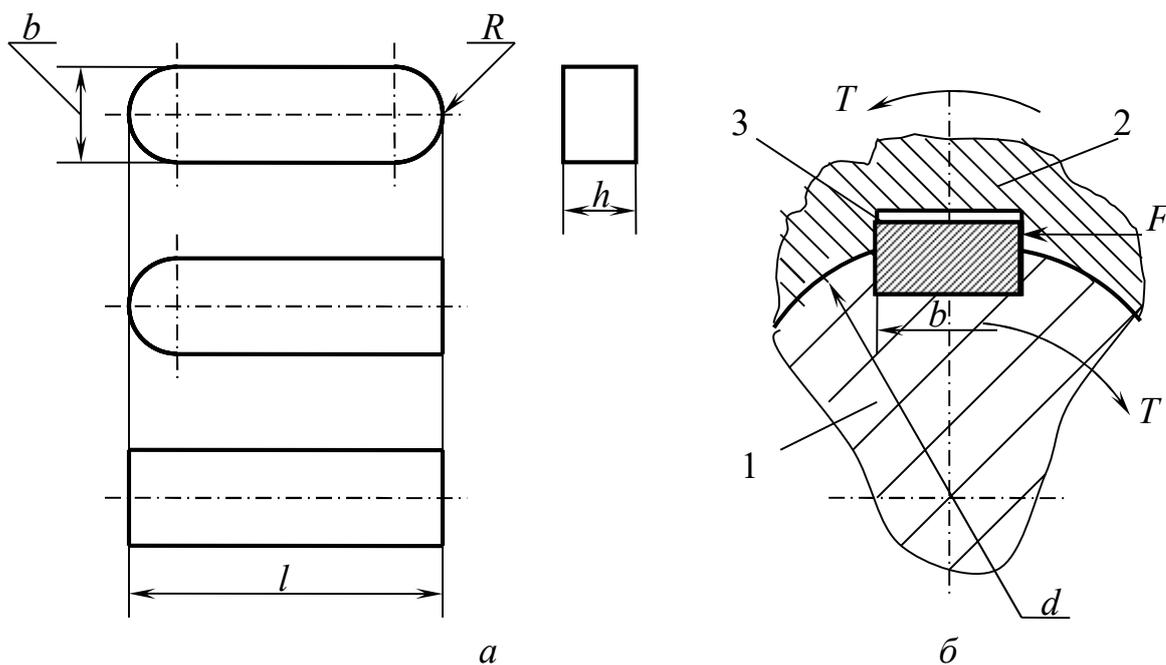


Рис. 1.1. Соединение призматической шпонкой

На валу 1 и во втулке 2 выполняются пазы, в которые при сборке устанавливается шпонка 3 (рис. 1.1, б), и крутящий момент T передается с вала к втулке ее боковыми гранями.

Боковые грани шпонки испытывают нормальные напряжения смятия, а в ее продольном сечении действуют касательные напряжения среза, но соотношение размеров стандартной низкой шпонки таково, что проверку на прочность по касательным напряжениям можно не выполнять. Таким образом, расчет соединения на прочность производится по напряжениям смятия, которые равны

$$\sigma_{\text{см}} = \frac{F}{A} = \frac{4T}{dhl_p} \leq [\sigma_{\text{см}}] \quad (1.1)$$

где F – сила, действующая на боковую грань; A – рабочий участок боковой грани, по которому распределяется сила F ; l_p – рабочая длина шпонки; $[\sigma_{см}]$ – допускаемое напряжение смятия.

Формула (1.1) получена с учетом следующих допущений:

- плечо силы F относительно центра сечения вала равно $d/2$;
- ширина участка боковой грани, на котором распределяется сила F , равна $h/2$.

Рабочая длина шпонки со скругленными торцами равна

$$l_p = l - b; \quad (1.2)$$

с одним скругленным торцом

$$l_p = l - b/2; \quad (1.3)$$

с плоскими торцами $l_p = l$.

Допускаемые напряжения $[\sigma_{см}] = 80 \dots 120$ МПа для соединений с переходными посадками втулки на вал, а для соединений с посадкой с натягом $[\sigma_{см}] = 110 \dots 200$ МПа. Меньшие напряжения принимают при чугунной втулке или при резких изменениях нагрузки, бóльшие – при стальной втулке и спокойной нагрузке.

1.3. Зубчатые (шлицевые) соединения

Соединения образуются сопряжением наружных зубьев на валу с внутренними зубьями в отверстии втулки.

По форме профиля зубьев различают соединения **прямобочные** (рис. 1.2, а), **эвольвентные** (рис. 1.2, б) и **треугольные** (рис. 1.2, в).

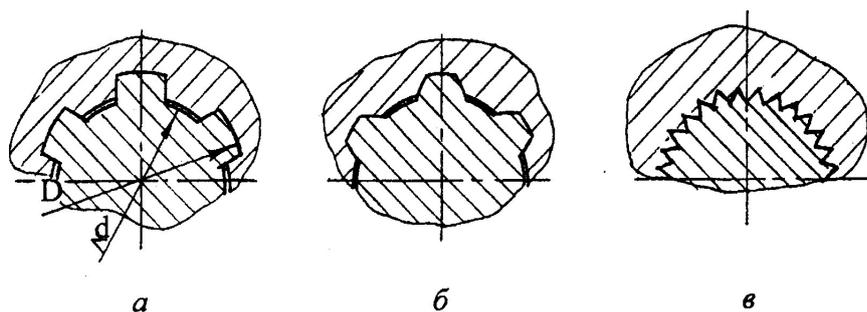


Рис. 1.2. Разновидности шлицевых соединений по форме профиля зуба

Соединения прямобочные выполняют с **центрированием** или по боковым сторонам зубьев, или по наружному диаметру, или по внутреннему диаметру вала. Центрирование по диаметрам обеспечивает более точную соосность соединяемых деталей, а центрирование по боковым сторонам – более точное

распределение нагрузки между зубьями, т.е. передачу бóльших крутящих моментов. На рис. 1.2, а показано центрирование по наружному диаметру.

Эвольвентные соединения выполняют с центрированием или по боковым сторонам зубьев, или по наружному диаметру. На рис. 1.2, б показано центрирование по наружному диаметру. Эвольвентные шлицы можно получать на зуборезном оборудовании и достигать при этом высокой точности.

Размеры прямобочных соединений даны в ГОСТ 1139, эвольвентных – в ГОСТ 6033.

Треугольные шлицы не стандартизованы. Они применяются при тонкостенных втулках, а также для соединения пластмассовых деталей с металлическими валами.

С точки зрения применения в ММ и роботах шлицевые соединения обладают такими преимуществами перед шпоночными, как повышенные нагрузочная способность и точность центрирования деталей. К недостаткам можно отнести сложность изготовления.

Основными критериями работоспособности шлицевых соединений являются прочность на смятие и сопротивление коррозионно-механическому износу. Причина такого изнашивания в неподвижных соединениях заключается в микро-перемещениях сопряженных поверхностей при вращении вала.

При допущении равномерного распределения нагрузки между зубьями условие прочности по напряжениям смятия выглядит следующим образом:

$$\sigma_{\text{см}} = \frac{2T}{zhd_m l} \leq [\sigma_{\text{см}}], \quad (1.4)$$

где z – число зубьев; h – высота зуба; d_m – средний диаметр соединения; l – длина поверхности контакта.

Высота зуба и средний диаметр определяют по формулам

$$h = (D - d)/2 - f'; \quad (1.5)$$

$$d_m = (D + d)/2, \quad (1.6)$$

где D , d – диаметры вершин и впадин зубьев соответственно; f' – расчетный зазор в соединении.

Допускаемое напряжение смятия определяют с учетом условий эксплуатации и твердости зубьев по табл. 1.1.

В случае постоянной нагрузки с числом циклов нагружения за полный срок службы порядка 10^8 условие удовлетворительного сопротивления соединения изнашиванию выражается неравенством

$$\sigma_{\text{см}} \leq [\sigma_{\text{изн}}]. \quad (1.7)$$

Значения $[\sigma_{см}]$ шлицевых соединений

Условия эксплуатации	$[\sigma_{см}]$, МПа	
	Твердость зубьев <i>HB350</i>	Твердость зубьев <i>HRC40</i>
Тяжелые (с ударом)	26...38	30...52
Средние	45...75	75...105
Легкие	60...90	90...150

Условное допускаемое напряжение $[\sigma_{изн}]$ изменяется в широких пределах в зависимости от твердости поверхностей контакта и условий приложения нагрузки. Для улучшенных зубьев $[\sigma_{изн}] = 26...85$ МПа; для закаленных: до *HRC40* $[\sigma_{изн}] = 34...105$ МПа; до *HRC45* $[\sigma_{изн}] = 42...130$ МПа. При необходимости точного определения $[\sigma_{см}]$, $[\sigma_{изн}]$ следует обратиться к литературе [3].

1.4. Резьбовые соединения

1.4.1. Классификация резьбы

По форме основной поверхности различают **цилиндрические** и **конические** резьбы. Наиболее распространена цилиндрическая резьба. Коническую применяют для плотных соединений труб, масленок, пробок и т.п.

По форме профиля резьбового выступа различают треугольные, круглые, прямоугольные, трапецеидальные и упорные резьбы. Форма профиля тесно связана с назначением резьбы: для образования соединений используются треугольные и круглые (**крепежные**) резьбы, а в винтовых механизмах – прямоугольные, трапецеидальные и упорные (**ходовые**) резьбы. Такое распределение объясняется более высокой относительной прочностью крепежной резьбы и большими силами трения в соединении крепежной резьбой.

По направлению винтовой линии различают **правую** и **левую** резьбу. У правой резьбы винтовая линия идет слева направо и вверх, у левой – справа налево и вверх. Обычно применяют правую резьбу, левую – только в некоторых специальных случаях.

Треугольные резьбы делятся на **метрические** и **дюймовые**. Геометрические параметры метрических резьб выражены в миллиметрах, дюймовых – в долях дюйма.

Разновидность дюймовой резьбы – **трубная** резьба, резьбовые выступы и впадины которой скруглены. Соединения трубными резьбами имеют меньшие зазоры, чем соединения метрическими резьбами, поэтому применяются в трубопроводной арматуре.

В ММ и роботах преимущественно применяются треугольные резьбы, они и будут рассмотрены в дальнейшем.

1.4.2. Геометрические параметры треугольной резьбы

Основные геометрические параметры (рис. 1.3): α – угол профиля, для метрической резьбы $\alpha = 60^\circ$, для дюймовой резьбы $\alpha = 55^\circ$; d – наружный диаметр; d_1 – внутренний диаметр; d_2 – средний диаметр; p – шаг резьбы.

Шаг резьбы – расстояние между одноименными сторонами соседних профилей, измеренное в направлении оси резьбы. По величине шага различают резьбы с крупным шагом и с мелкими шагами. Крупный шаг для определенного d один, а мелких шагов несколько. С уменьшением шага соответственно уменьшаются размеры резьбового выступа и угол подъема витка (см. ниже).

По образующей воображаемого цилиндра, диаметр которого равен среднему диаметру резьбы, ширина резьбового выступа равна ширине впадины (и равна $b/2$).

Кроме того, выделяют такие параметры, как n – число заходов; $p_1 = np$ – ход резьбы, т.е. перемещение гайки по винту за один оборот; ψ – угол подъема витка.

Под углом ψ подразумевается угол подъема развертки винтовой линии по среднему диаметру:

$$\psi = \operatorname{arctg} \frac{p_1}{\pi d_2} = \operatorname{arctg} \frac{np}{\pi d_2}. \quad (1.8)$$

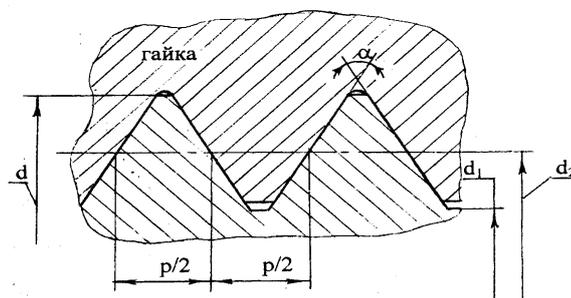


Рис. 1.3. Основные геометрические параметры треугольной резьбы

От величины ψ зависит, будет ли резьба **самотормозящейся**. Самоторможение – неременное условие для крепежной резьбы, поскольку без его соблюдения соединение не в состоянии выдерживать осевую нагрузку. В однозаходной треугольной резьбе $\psi = 2^\circ 30' \dots 3^\circ 30'$, что дает гарантированное самоторможение.

1.4.3. Предотвращение самоотвинчивания в резьбовых соединениях

Весьма часто резьбовые соединения эксплуатируются в условиях вибрации, переменных и ударных нагрузок. При этом обеспечения условия

самоторможения недостаточно для предотвращения самоотвинчивания, т.к. вследствие переменного характера нагрузки силы трения в резьбе понижаются.

Существует много способов дополнительного **стопорения** резьбы [1]. Способы первой группы направлены на повышение и стабилизацию сил трения в резьбе. Основные и наиболее часто применяемые из них – постановка **контргайки** и **пружинной шайбы**. Контргайка создает дополнительное натяжение, а, следовательно, и трение в резьбе, не зависящее от внешней нагрузки. Пружинная шайба представляет собой, по сути дела, виток пружины и поддерживает натяг и трение в резьбе на участке самоотвинчивания в один – два оборота гайки.

Способы второй группы сводятся к жесткому креплению элементов (гайки с болтом, гайки или винта с деталью). Одним из таких способов является применение в соединении специальной **корончатой гайки** со **шплинтом**. Корончатая гайка имеет кольцевой выступ с прорезями (коронку). Шплинтом называется деталь, изготовленная из проволоки полукруглого сечения. После навинчивания гайки на резьбовый стержень шплинт вставляется в прорезь коронки так, что проходит через коронку и резьбовый стержень насквозь (в стержне заранее сделано отверстие под шплинт). Затем концы шплинта отгибают, после чего шплинт надежно фиксирует гайку относительно резьбового стержня.

Указанными способами можно предотвратить самоотвинчивание в большинстве резьбовых соединений. В противном случае следует обратиться к литературе и подобрать приемлемый способ.

1.4.4. Расчет резьбовых соединений на прочность

Основные виды разрушения резьбовых соединений – срез витков и разрыв резьбового стержня. Касательные напряжения среза зависят, при равных фиксированных диаметре и шаге резьбы, от количества витков резьбы, по которым распределяется нагрузка, т.е. от высоты гайки. Поэтому добиться равнопрочности резьбы и стержня винта можно подбором высоты гайки. Стандартная высота нормальной гайки $H \approx 0,8d$, и именно такая высота дает выполнение условия равнопрочности. Следовательно, при использовании в болтовом соединении гайки нормальной высоты исключается необходимость рассмотрения прочности витков, и расчет соединения сводится к **расчету стержня болта (винта)**.

Ниже рассмотрены распространенные в конструкциях ММ и роботов случаи нагружения резьбового стержня.

Случай 1. Стержень винта нагружен только внешней растягивающей силой F , затяжка соединения отсутствует.

Опасное сечение – по внутреннему диаметру резьбы. Условие прочности по напряжениям растяжения в стержне

$$\sigma = \frac{F}{\frac{\pi d_1^2}{4}} = \frac{4F}{\pi d_1^2} \leq [\sigma]. \quad (1.9)$$

Допускаемые напряжения $[\sigma]$ здесь и далее см. в табл. 1.2.

Случай 2. Внешняя нагрузка отсутствует, соединение затянуто.

Случай характерен для крепления ненагруженных герметичных крышек, люков и т.п.

Стержень болта растягивается осевой **силой затяжки** $F_{\text{зат}}$ и закручивается **моментом сил** T , необходимым для обеспечения затяжки. Величина $F_{\text{зат}}$ определяется из условия герметичности по рекомендациям, учитывающим опыт эксплуатации аналогичных соединений. Расчет стержня производится по **эквивалентному напряжению**, учитывающему наличие как нормальных напряжений растяжения, так и касательных напряжений кручения. Для стандартной метрической резьбы соотношение эквивалентного и нормального напряжений выражается зависимостью $\sigma_3 \approx 1,3\sigma$, что позволяет рассчитывать стержень болта по формуле

$$\sigma_3 = \frac{1,3F_{\text{зат}}}{\frac{\pi d_1^2}{4}} = \frac{5,2F_{\text{зат}}}{\pi d_1^2} \leq [\sigma]. \quad (1.10)$$

Случай 3. Соединение нагружено силами, сдвигающими детали в стыке, болт поставлен в отверстии с зазором. Пример – соединение, показанное на рис. 1.4.

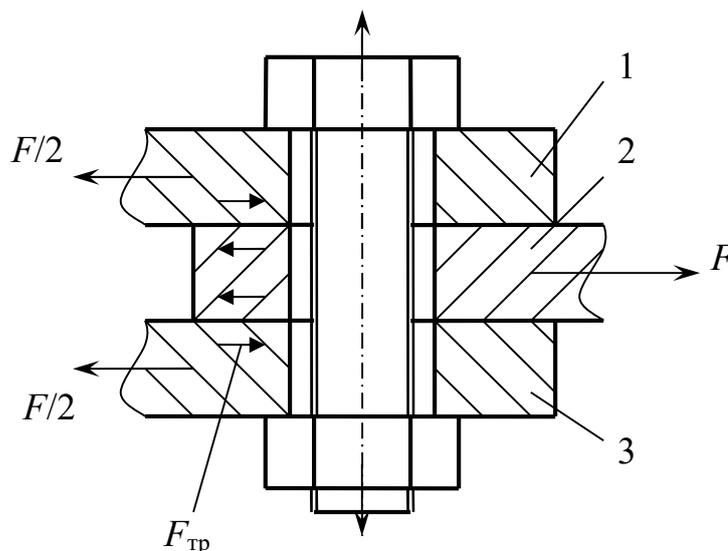


Рис. 1.4. Соединение, нагруженное поперечными силами (болт в отверстии с зазором)

Для упрощения расчета принято следующее допущение: болт не касается своей боковой поверхностью стенок отверстий в деталях. Следовательно, сдвигу деталей препятствуют только силы трения в стыке. Условие отсутствия сдвига может быть получено из рассмотрения равновесия детали 2:

$$F = iF_{\text{тр}} = iF_{\text{зат}}f, \quad (1.11)$$

где i – число плоскостей стыка деталей; f – коэффициент трения скольжения в стыке (для стальных деталей $f = 0,15 \dots 0,2$).

Требованиям практики равенство (1.11) не удовлетворяет, т.к. малейшее увеличение силы F или уменьшение коэффициента трения (например, в результате попадания смазки в соединение) приведет к сдвигу деталей. Поэтому вместо (1.11) используется выражение

$$KF = iF_{\text{зат}}f, \quad (1.12)$$

откуда

$$F_{\text{зат}} = \frac{KF}{if}, \quad (1.13)$$

в (1.13) K – коэффициент запаса. При статической нагрузке $K = 1,3 \dots 1,5$, при переменной нагрузке $K = 1,8 \dots 2$.

После определения $F_{\text{зат}}$ прочность болта оценивают по (1.10).

Случай 4. Соединение нагружено силами, сдвигающими детали в стыке, болт поставлен без зазора. Такие соединения (см. рис. 1.5) образуются с помощью болтов по ГОСТ 7817, имеющих гладкую рабочую часть, диаметр которой d_3 больше диаметра резьбы d . Отверстия под болты обрабатывают разверткой, в результате посадки болтов в отверстиях получаются переходные или с натягом.

Сдвигающие силы вызывают в стержне болта напряжения среза и смятия. Резьба в соединении играет вспомогательную роль, фиксируя соединяемые детали одну относительно другой.

Напряжение среза в стержне болта в соединении по рис. 1.5

$$\tau = \frac{4F}{\pi d_3^2 i} \leq [\tau], \quad (1.14)$$

где i – число плоскостей среза (на рис. 1.5 $i = 2$).

Напряжение смятия для крайней детали

$$\sigma_{\text{см1}} = \frac{F}{2d_3\delta_1} \leq [\sigma_{\text{см}}]; \quad (1.15)$$

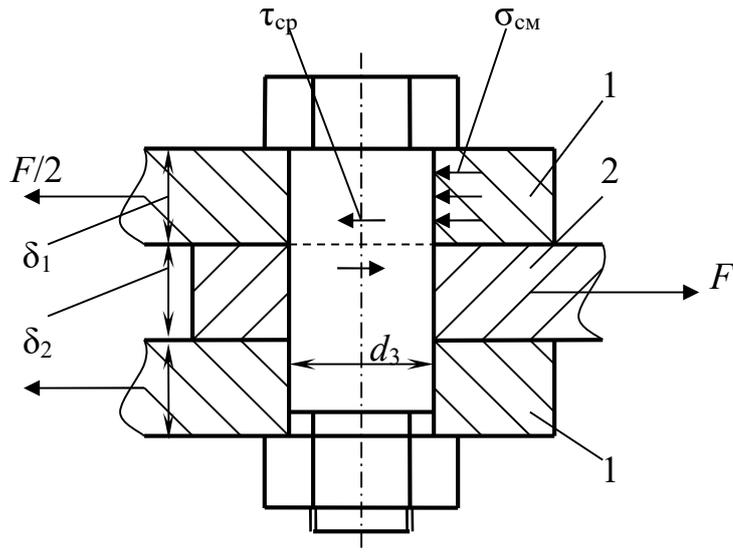


Рис. 1.5. Соединение, нагруженное поперечными силами (болт в отверстии без зазора)

для средней детали

$$\sigma_{см2} = \frac{F}{d_3 \delta_2} \leq [\sigma_{см}]. \quad (1.16)$$

Формулы (1.15), (1.16) справедливы как для болта, так и для деталей. Из двух значений $\sigma_{см}$ расчет прочности выполняют по наибольшему, а допускаемое напряжение определяют по более слабому материалу болта или детали. Обычно диаметр болта находят из условия прочности на срез, а затем производят проверку по напряжениям смятия.

Случай 5. Болт затянут, внешняя нагрузка раскрывает стык деталей. В качестве примера могут быть рассмотрены болты крепления крышки резервуара к корпусу (рис. 1.6).

Внутри резервуара находится газ под избыточным давлением p . Очевидно, что затяжка болтов должна обеспечивать герметичность соединения, для чего болты предварительно (до того, как в резервуар подается газ) затягивают. При этом болты и стык деформируются: болты растягиваются, стык сжимается. Сжатие стыка происходит в основном за счет прокладки, если предусмотрена мягкая прокладка. Если же прокладка металлическая, то главную роль играет податливость фланцев крышки и корпуса. После того, как в резервуаре установилось давление p , приходящаяся на болт внешняя нагрузка становится равной

$$F = \frac{p\pi D^2}{4z}, \quad (1.17)$$

где z – число болтов.

Под действием внешней нагрузки дополнительно растягиваются. Но при этом крышка приподнимается болты, и сжатие стыка уменьшается на величину дополнительной деформации болтов. Таким образом, с одной стороны, имеет место приращение нагрузки на болт за счет силы давления газа на крышку, а с другой стороны, уменьшается нагрузка на болт со стороны стыка, возникающая в результате предварительной затяжки. В итоге суммарное увеличение нагрузки на болт оказывается значительно меньше, чем F по (1.17).

Расчетная суммарная нагрузка на болт

$$F_p = F_{\text{зат}} + \chi F, \quad (1.18)$$

где χ – коэффициент внешней нагрузки, обычно принимается равным 0,2...0,3.

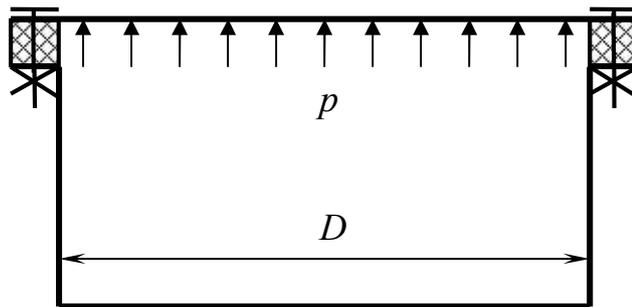


Рис. 1.6. Соединение «болт затянут, внешняя нагрузка раскрывает стык»

Силу затяжки рекомендуется принимать

$$F_{\text{зат}} = k_{\text{зат}} F, \quad (1.19)$$

где $k_{\text{зат}}$ – коэффициент затяжки.

По условию герметичности: при мягкой прокладке $k_{\text{зат}} = 1,3...2,5$; при металлической фасонной прокладке $k_{\text{зат}} = 2...3,5$; при металлической плоской прокладке $k_{\text{зат}} = 3...5$.

После того, как найдена F_p , проверяют болт на прочность по формуле

$$\sigma_s = \frac{5,2F_p}{\pi d_1^2} \leq [\sigma]. \quad (1.20)$$

Случай 6. Эксцентрично нагруженный болт. Пример – нагружение болта с молотовидной головкой (рис. 1.7). Такие болты используют, когда невозможно расположить в отверстии обычный болт (отверстие слишком близко к стенке), а также в некоторых других случаях.

Затяжка соединения вызывает возникновение в стержне болта напряжений растяжения

$$\sigma_p = \frac{4F_{\text{зат}}}{\pi d_1^2} \quad (1.21)$$

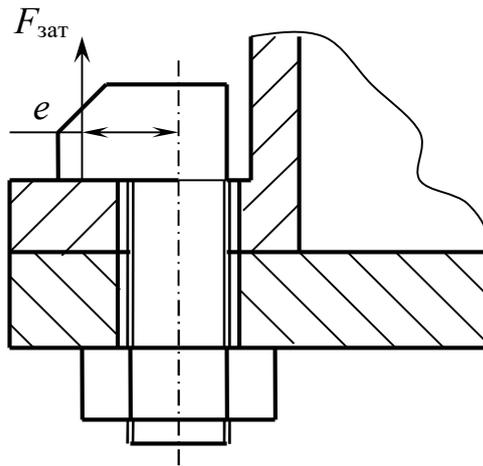


Рис. 1.7. Соединение болтом с молотовидной головкой

и напряжений изгиба

$$\sigma_{\text{и}} = \frac{32F_{\text{зат}}e}{\pi d_1^3}, \quad (1.22)$$

где e – эксцентриситет силы затяжки.

Для сопоставления величин составляющих напряжений предположим, что $e = d_1$. Тогда

$$\sigma_{\text{и}} = \frac{32F_{\text{зат}}}{\pi d_1^2}; \quad (1.23)$$

$$\sigma_{\text{и}} / \sigma_p = 8. \quad (1.24)$$

Из (1.24) видно, что из двух составляющих гораздо более опасно напряжение изгиба. Следовательно, эксцентричного нагружения болтов нужно всемерно избегать, а в тех случаях, когда использование эксцентрично

нагруженного болта является технической необходимостью, обязательно учитывать его в расчетах.

Суммарное напряжение в стержне болта с учетом напряжения кручения

$$\sigma_p = \frac{5,2F_{\text{зат}}}{\pi d_1^2} + \frac{32F_{\text{зат}}e}{\pi d_1^3} \leq [\sigma]. \quad (1.25)$$

1.4.5. Материалы и допускаемые напряжения

Стандартные крепежные изделия изготавливают обычно из сталей марок Сталь 10...Сталь 35, т.к. эти дешевые стали позволяют выпускать большие количества изделий наиболее производительными методами. Стали с более высокими прочностными характеристиками применяют для изготовления высоконагруженных деталей в ответственных соединениях. В этих случаях может предусматриваться также термическая обработка.

Особое внимание следует уделять защите соединений от коррозии. С этой целью стандарты предусматривают больше десятка различных видов покрытий болтов, шпилек и гаек применительно к различным агрессивным средам – от цинкового с хромированием до серебряного. В тех случаях, когда вид среды неизвестен, можно рекомендовать достаточно простые и дешевые покрытия, такие, как кадмиевое с хромированием (группа 02).

В технически обоснованных случаях крепежные детали выполняют из цветных металлов и сплавов.

Допускаемые напряжения при расчете резьбовых соединений на прочность сведены в табл. 1.2.

Таблица 1.2

Допускаемые напряжения

Случай соединения	Номер формулы	Значение допускаемого напряжения
1	(1.9)	$[\sigma] = 0,6\sigma_T$
2,3,5	(1.10), (1.20)	$[\sigma] = \sigma_T/[s]$; [s] – по табл. 1.3 для неконтролируемой затяжки; [s] = 1,5...2,5 – для контролируемой затяжки
4	(1.14), (1.15), (1.16)	$[\tau] = 0,4\sigma_T$ – для статической нагрузки; $[\tau] = (0,2...0,3)\sigma_T$ – для переменной нагрузки; $[\sigma_{см}] = 0,8\sigma_T$ – сталь; $[\sigma_{см}] = (0,4...0,5)\sigma_T$ – чугун
6	(1.25)	$[\sigma] = 0,6\sigma_T$

Различают затяжку **контролируемую** и **неконтролируемую**. Контролируемая затяжка осуществляется с помощью специальных динамометрических ключей и ключей предельного момента, дающих

возможность затянуть соединение заданной силой (и не большей). Существуют также и другие средства и методы контроля затяжки, к которым следует прибегать там, где это оговорено техническими требованиями. Судя по величинам запасов прочности (табл. 1.3), контролируемая затяжка позволяет существенно повысить надежность соединений.

1.5. Заклепочные соединения

1.5.1. Разновидности заклепочных соединений

Заклепочное соединение образуется расклепыванием стержня заклепки, вставленной в отверстия деталей (рис. 1.8). Обжимка 1 формирует замыкающую головку 2 заклепки 3, причем вследствие пластических деформаций стержень заклепки заполняет зазор в отверстиях. Поддержка 4 фиксирует закладную головку 5 заклепки.

Силы, вызванные упругими деформациями деталей и стержня заклепки, стягивают детали. Сдвигу деталей препятствует сопротивление стержня заклепки и частично силы трения между деталями.

Таблица 1.3

Запасы прочности при неконтролируемой затяжке

Материал болта	Запас прочности [s] для резьбы		
	M6...M16	M16...M30	M30...M60
Углеродистая сталь	5...4	4...2,5	2,5...1,5
Легированная сталь	6,5...5	5...3,3	3,3

Отверстия в деталях сверлят или продавливают. Сверление менее производительно, но придает соединению повышенную прочность. В ответственных соединениях предусматривается обязательное совместное сверление отверстий в деталях, что дополнительно повышает надежность соединения.

Клепку можно производить вручную и машинным способом (пневматическими молотками, прессами и т.п.).

Стальные заклепки диаметром до 10 мм и заклепки из цветных металлов ставят без нагрева, поэтому процесс расклепывания называют **холодной клепкой**. Стальные заклепки большого диаметра ставят с нагревом. Нагрев повышает пластичность заклепки, облегчает расклепывание, улучшает заполнение отверстия и повышает натяг в соединении, связанный с температурными деформациями при остывании. В этом случае образование соединения называют **горячей клепкой**.

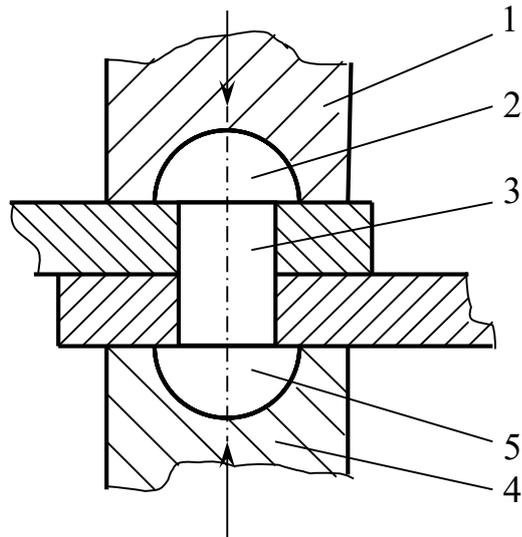


Рис. 1.8. Заклепочное соединение

Применяются заклепки с полукруглой головкой (такая заклепка показана на рис. 1.8), с потайной и полупотайной головками. Кроме того, существуют различные типы специальных заклепок: пустотелые и полупустотелые, заклепки для односторонней клепки и т.д. Геометрическая форма и размеры заклепок нормальной точности оговариваются ГОСТ 10299, ГОСТ 10300, заклепок повышенной точности – ГОСТ 14787, ГОСТ 14798, ГОСТ 14801.

Листовые детали соединяются заклепочными **швами**. В зависимости от назначения различают швы **прочные, плотные и прочноплотные**. Прочные швы применяют в металлоконструкциях, плотные – в резервуарах для хранения жидкостей и газов с невысоким давлением, прочноплотные – в резервуарах для хранения жидкостей и газов с высоким давлением. Плотность шва достигается постановкой заклепок с шагом, не большим некоторого строго определенного значения.

По конструктивному признаку различают швы **однорядные и многорядные**, соединения **внахлестку и встык, односрезные и многосрезные**. На рис. 1.9 приведен пример двухсрезного соединения внахлестку.

Применение заклепочного соединения целесообразно в тех случаях, когда материалы деталей плохо соединяются сваркой, а также в тех конструкциях, где важно растянуть во времени процесс разрушения.

К недостаткам соединения относятся трудоемкость выполнения длинных заклепочных швов, вредность работы клепальщика, существенное ослабление соединяемых деталей отверстиями под заклепки.

1.5.2. Расчет заклепочного соединения на прочность

На основные размеры заклепочных соединений выработаны нормы, по которым выбирают диаметры отверстия и заклепки, шаг шва и расстояние от

шва до края деталей, а также толщину деталей. Расчет заклепки обычно носит проверочный характер.

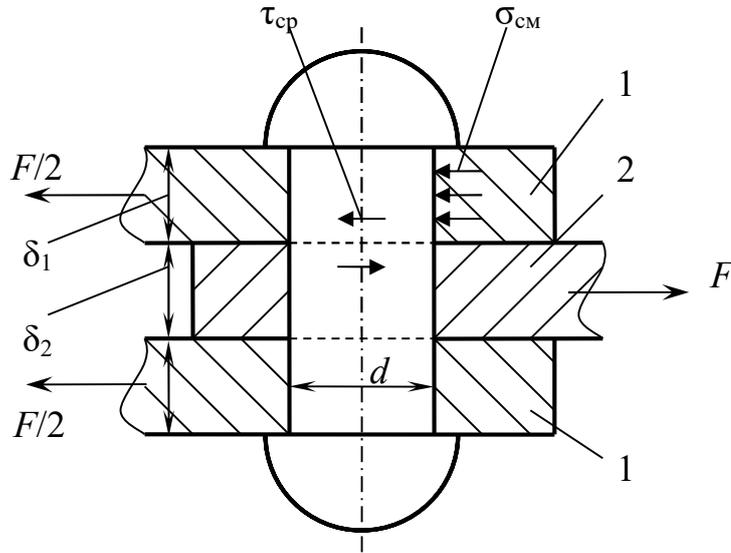


Рис. 1.9. Двухсрезное заклепочное соединение внахлестку

Рассмотрим соединение, нагруженное силами, сдвигающими детали в стыке (рис. 1.9). Его расчет аналогичен приведенному выше расчету резьбового соединения болтом, поставленным в отверстия без зазора, поэтому дополнительных пояснений не требует.

Условие прочности заклепки по напряжениям среза выражается формулой

$$\tau = \frac{4F}{\pi d^2 i} \leq [\tau_{\text{ср}}], \quad (1.26)$$

условия прочности по напряжениям смятия:

$$\sigma_{\text{см1}} = \frac{F}{2d \delta_1} \leq [\sigma_{\text{см}}], \quad (1.27)$$

$$\sigma_{\text{см2}} = \frac{F}{d \delta_2} \leq [\sigma_{\text{см}}]. \quad (1.28)$$

Допускаемые напряжения для заклепки из стали Ст3 принимают такими: $[\tau_{\text{ср}}] = 140$ (100) МПа; $[\sigma_{\text{см}}] = 320$ (280) МПа. Первые числа – для отверстий, полученных сверлением, значения в скобках – для отверстий, полученных продавливанием.

Материал заклепки должен отвечать следующим требованиям:

– обладать пластичностью;

- не принимать закалки при горячей клепке;
- не образовывать с материалом деталей гальванической пары.

1.6. Сварные соединения

1.6.1. Виды сварки

Из всего многообразия применяемых в настоящее время в производстве видов сварки при изготовлении ММ и роботов преимущественно используются **электродуговая и контактная**.

В электродуговой сварке образование соединения основано на заполнении стыка между деталями металлом **электрода**, расплавленного электрической дугой. Данный способ требует качественного прогрева кромок деталей и предотвращения доступа в зону сварки кислорода и азота воздуха. Последнее обеспечивается специальным покрытием электрода, которое, разлагаясь под действием температуры дуги, выделяет большое количество газа, инертного по отношению к металлу.

Различают **ручную и автоматическую** электродуговую сварку. Шов, выполненный сварочным автоматом, имеет более высокое качество, а следовательно, и большую статическую и усталостную прочность. К сожалению, выполнить соединение автоматически далеко не всегда возможно.

Контактная сварка является высокопроизводительным методом и применяется для соединения листовых деталей толщиной до 4 мм. Она основана на использовании **повышенного электрического сопротивления** зоны контакта деталей. Различают **точечную, шовную и стыковую** контактную сварку.

Сущность контактной сварки удобно пояснить на примере ее точечной разновидности (рис. 1.10).

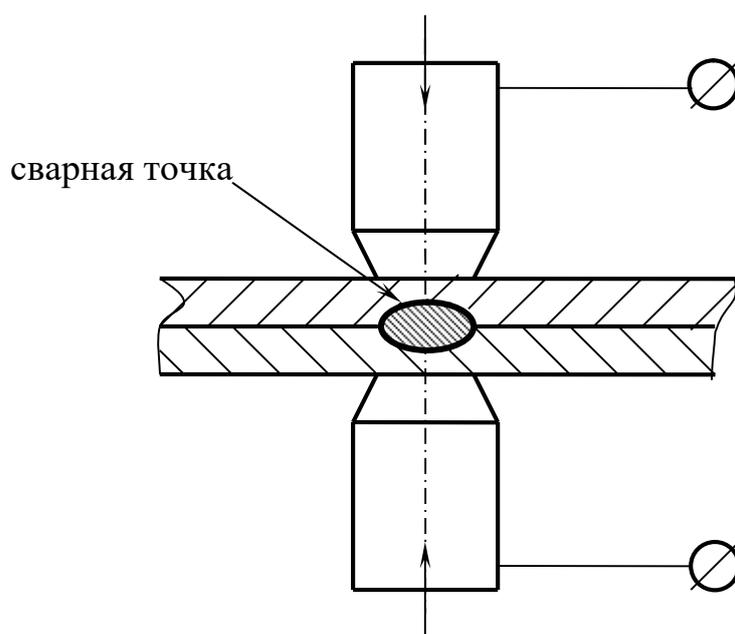


Рис. 1.10. Схема контактной точечной сварки

Детали сжимаются электродами. Ток течет между электродами, при этом теплота в основном выделяется на поверхности контакта деталей; металл плавится, и образуется сварная точка.

Шовная сварка выполняется аналогично, но в качестве электродов применяют диски, которые перекатываются по деталям в направлении шва. Появляется возможность провести герметичный шов.

Стыковая контактная сварка применяется для соединения встык деталей типа стержней со сравнительно небольшой площадью поперечного сечения.

Достоинства сварного соединения следующие:

- высокая производительность и сравнительно невысокая трудоемкость сварки;
- простота обеспечения равнопрочности изделия, снижение его массы и стоимости.

Недостатки:

- необходимость правильного выбора материалов деталей;
- наличие в шве дефектов (неоднородностей, микротрещин и т.п.), и, как следствие, снижение прочности соединения.

Лучше всего свариваются детали из низкоуглеродистых сталей, например, из стали Ст3. Стали углеродистые и легированные требуют для сварки применения специальных приемов: предварительного прогрева деталей, подачи инертного газа в зону сварки и т.д.

1.6.2. Соединения ручной электродуговой сваркой

Ручная электродуговая сварка представляет собой наиболее универсальный способ образования соединений, поэтому именно она и будет рассмотрена подробно.

Элементы сварных швов, получаемых ручной электродуговой сваркой, указаны в ГОСТ 5264. Стандарт устанавливает четыре типа соединений в зависимости от взаимного расположения соединяемых деталей: **стыковое, нахлесточное, тавровое и угловое.**

Стыковое соединение (рис. 1.11) простое и зачастую наиболее надежное. При толщине деталей $s \leq 6$ мм их можно соединять без **разделки кромок** (соединение С2). В случае $s > 6$ мм расплавленный металл электрода не может заполнить зазор между деталями, получается некачественный шов пониженной прочности. Поэтому при толщинах больших 6 мм применяют подварку с другой стороны, одностороннюю и двухстороннюю разделку кромок деталей (например, соединения С5 и С21).

Нахлесточное соединение (рис. 1.12) возникло как аналог заклепочного соединения внахлестку. Из всех сварных соединений оно наиболее простое, не требует подготовки кромок независимо от толщины деталей. Возможны одностороннее (Н1) и двухстороннее (Н2) нахлесточные соединения.

Тавровое соединение показано на рис. 1.13 и, подобно нахлесточному, может быть односторонним и двухсторонним. Кроме того, при больших s оно выполняется с разделкой кромок пристыковываемой детали.

Угловое соединение (рис. 1.14) часто применяется при изготовлении различного рода металлических емкостей – корпусов, коробов и т.п.

Различают два вида **швов**: **стыковой шов** – для образования стыковых соединений; **угловой шов** – для всех остальных соединений.

В обозначение типа **электрода** для ручной электродуговой сварки по ГОСТ 9467 входит буква «Э» и число, равное пределу прочности металла электрода, выраженному в кгс/мм², например, Э42, Э50А. Буква «А» в обозначении показывает, что химический состав электрода подвергается дополнительному контролю. Такие электроды применяются в ответственных соединениях для повышения надежности конструкции.

Стандарт устанавливает ряд диаметров электродов в миллиметрах: 2, 3, 4, 5, 6, 8, 10, 12... Для ручной сварки используют электроды меньших диаметров, как правило, до 6 – 8 мм, т.к. при этом достигается наивысшее качество шва в сочетании с невысокой трудоемкостью сварки.

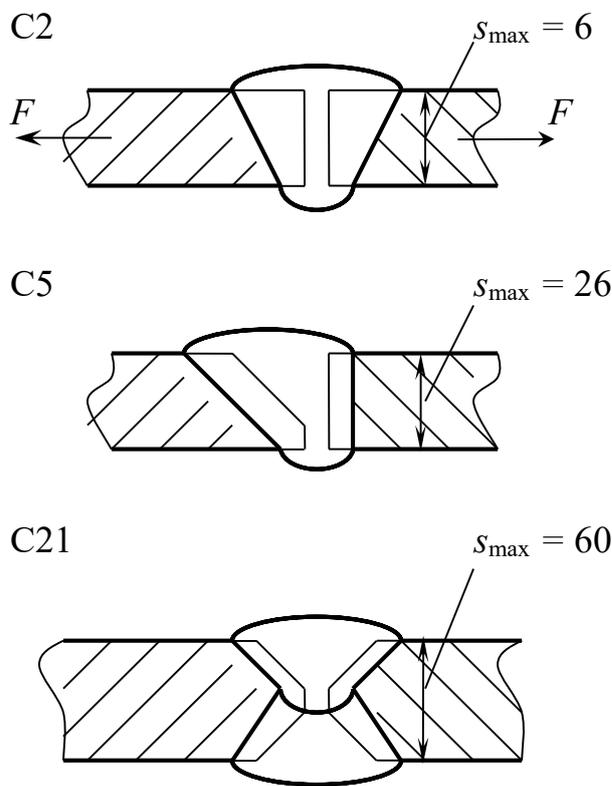


Рис. 1.11. Стыковое сварное соединение

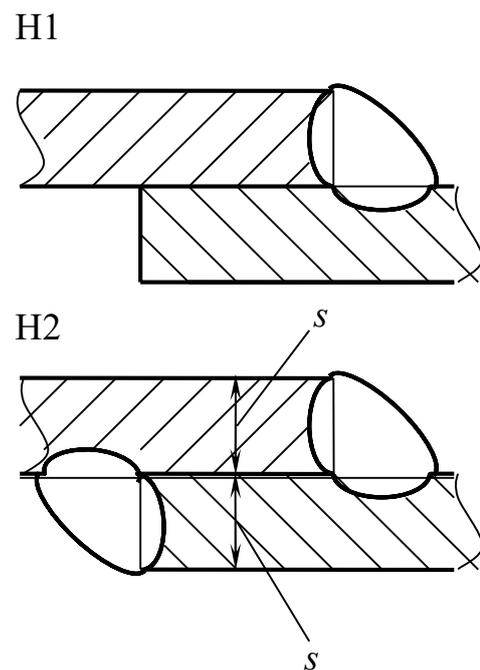


Рис. 1.12. Нахлесточное сварное соединение

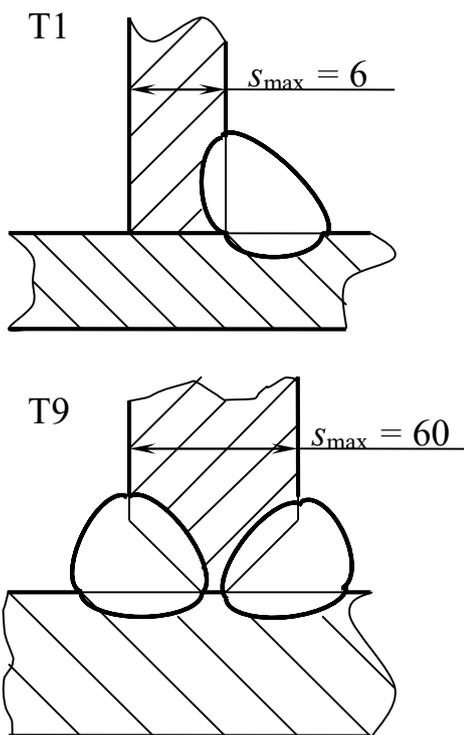


Рис. 1.13. Тавровое сварное соедление

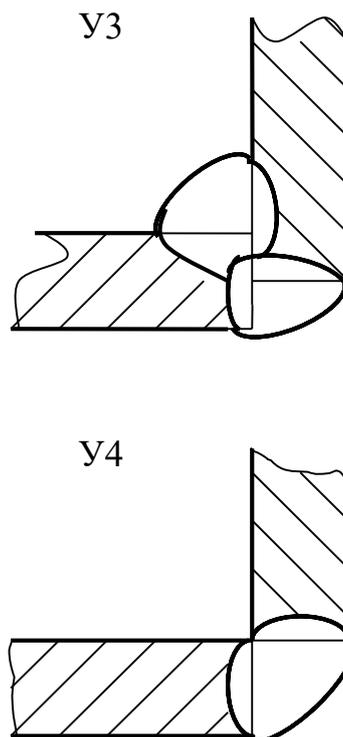


Рис. 1.14. Угловое сварное соедление

1.6.3. Расчет сварных соединений на прочность

Расчет **стыкового шва** производится следующим образом.

Напряжение в шве от растягивающей нагрузки (см. рис. 1.11) определяют по формуле

$$\sigma = \frac{F}{bs} \leq [\sigma'] = (0,9...1,0)[\sigma_p], \quad (1.29)$$

где b – длина шва; $[\sigma']$ – допускаемое напряжение для материала шва; $[\sigma_p]$ – допускаемое напряжение растяжения для материала деталей.

Предполагается, что стыковой шов практически равнопрочен с соединяемыми деталями. В (1.29) коэффициент 0,9 принимают при электродах Э42, Э50, а коэффициент 1,0 – при электродах Э42А, Э50А.

Напряженное состояние **углового шва** в нахлесточном и тавровом соединениях существенно отличается от напряженного состояния стыкового шва даже при простейшем нагружении растягивающими силами. В материале шва возникают как нормальные, так и касательные напряжения. Инженерный расчет производится упрощенно по касательным напряжениям. По форме швы разделяют на нормальные – в виде равнобедренного прямоугольного треугольника, выпуклые и вогнутые. Вогнутые швы лучше сопротивляются переменной нагрузке, но выполнение их связано с дополнительной

механической обработкой, а следовательно, и с дополнительными затратами. В дальнейшем рассматриваются **нормальные швы** как самые распространенные в практике.

На рис. 1.15 показано нахлесточное соединение нормальным угловым швом с длиной L и катетом K . Разрушение такого шва происходит по биссектрисе AB прямого угла, что предсказано теорией и подтверждено практикой.

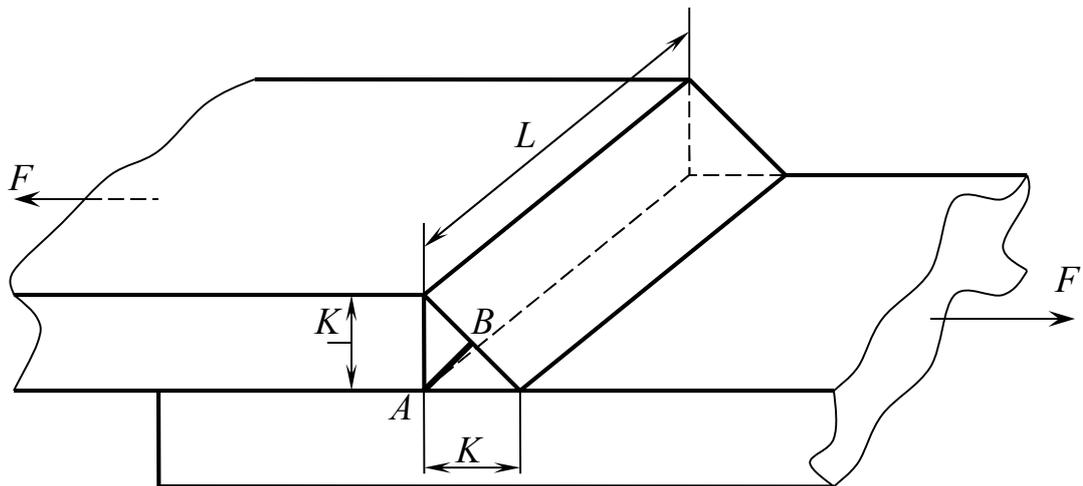


Рис. 1.15. Геометрия углового шва

Площадь опасного сечения шва

$$A_{o.c} = AB \cdot L \approx 0,7KL. \quad (1.30)$$

Условие прочности шва

$$\tau = \frac{F}{A_{o.c}} = \frac{F}{0,7KL} \leq [\tau']. \quad (1.31)$$

Допускаемое касательное напряжение для сварных швов, выполненных электродами Э42, Э50, принимают равным $[\tau'] = 0,6[\sigma_p]$, а для швов, выполненных электродами Э42А, Э50А – равным $[\tau'] = 0,65[\sigma_p]$.

По расположению относительно направления нагрузки различают швы **фланговые** (параллельные нагрузке), **лобовые** (перпендикулярные нагрузке) и **косые**. Напряжения в лобовом и фланговом швах различаются (при прочих равных условиях), но в инженерных расчетах касательные напряжения с достаточной степенью точности определяются по одним и тем же формулам.

На рис. 1.16 приведено соединение двумя фланговыми и одним лобовым швами. Для этого случая формула (1.31) принимает следующий вид:

$$\tau = \frac{F}{0,7K(2L_{\phi} + L_{\pi})} \leq [\tau'], \quad (1.32)$$

где L_{ϕ} , L_{π} – длины флангового и лобового швов.

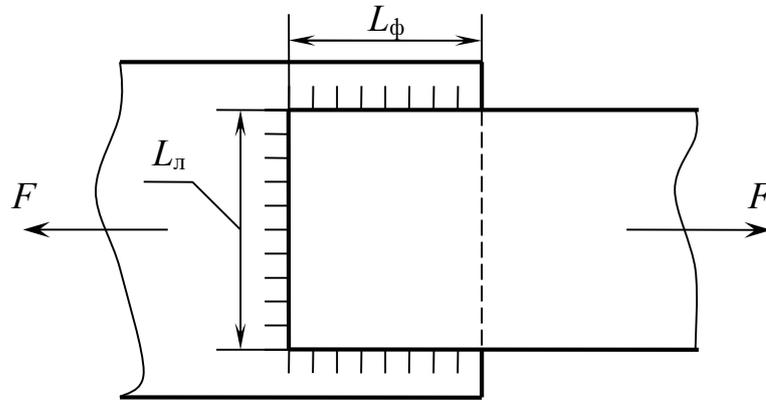


Рис. 1.16. Вариант нахлесточного соединения двумя фланговыми и одним лобовым швами

Соединения, показанные на рис. 1.17, нагруженные парой сил с моментом T , рассчитываются по следующим формулам:
соединение на рис. 1,17, *a*

$$\tau = \frac{T}{0,7KLb} \leq [\tau']; \quad (1.33)$$

на рис. 1.17, *б*

$$\tau = \frac{6T}{0,7Kb^2} \leq [\tau']; \quad (1.34)$$

на рис. 1.17, *в*

$$\tau = \frac{T}{0,7KLb + 0,7K \frac{b^2}{6}} \leq [\tau']. \quad (1.35)$$

В тех случаях, когда соединение находится одновременно под действием различных нагрузок (поперечная и продольная силы, крутящий момент и т.п.), расчет ведут, исходя из принципа независимости действия сил.

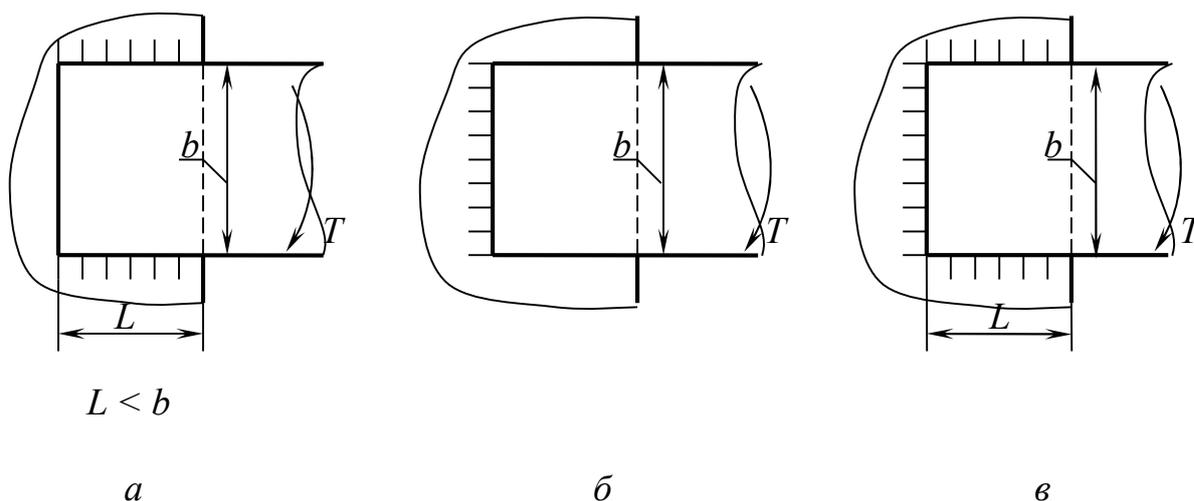


Рис. 1. 17. Соединения, нагруженные парой сил с моментом T

1.7. Соединения с натягом

Соединения типа «вал – ступица» с натягом (фрикционные) в ММ и роботах используются значительно чаще, чем в механизмах общего назначения, что объясняется их серьезными **преимуществами** по сравнению с прочими соединениями. Например, по сравнению со шпоночными соединения с натягом:

- имеют упрощенную технологию изготовления деталей;
- обеспечивают более точное базирование деталей в соединении;
- позволяют исключить элементы, снижающие прочность деталей, такие, как шпоночные пазы.

К **недостаткам** соединений с натягом относятся трудоемкость сборки и сложность контроля качества.

Основные способы сборки – сборка прессованием и сборка нагревом. При сборке нагревом нагревают ступицу (до температуры не выше $240\text{ }^{\circ}\text{C}$) или охлаждают вал (жидким воздухом до минус $190\text{ }^{\circ}\text{C}$, сухим льдом до минус $72\text{ }^{\circ}\text{C}$), что позволяет получать соединения, способные воспринимать бóльшие нагрузки, чем соединения прессованием.

Задача расчета соединения – правильный выбор посадки, натяг в которой обеспечит необходимое давление, а значит, и силы трения на поверхности контакта деталей. Методика расчета подробно рассмотрена в [1, 3].

2. ПЕРЕДАЧИ

2.1. Общие сведения

Механической передачей называется механизм, преобразующий параметры движения при его передаче от двигателя к исполнительным органам машины. Передача осуществляет согласование режима работы двигателя с режимом работы исполнительных органов. Применительно к ММ часто используется так-же термин **преобразователь движения**.

В ряде случаев передачи предназначены для изменения направления движения или для преобразования вращательного движения в поступательное или наоборот.

Часто в функцию передачи входит регулирование частоты вращения (скорости) исполнительного органа при постоянной скорости двигателя. Такая передача называется **вариатором**.

Основными параметрами движения являются **мощность** P_1 на входе и P_2 на выходе передачи и **частоты вращения** n_1 на входе и n_2 на выходе (либо **угловые скорости** ω_1 и ω_2 соответственно). Кроме того, различают производные характеристики:

– **коэффициент полезного действия** (КПД)

$$\eta = \frac{P_2}{P_1}, \quad (2.1)$$

– **передаточное отношение**, определяемое в направлении потока мощности,

$$i = n_1/n_2 = \omega_1/\omega_2. \quad (2.2)$$

По величине i передачи делятся на понижающие при $i > 1$ и $n_1 > n_2$, или **редукторы**, и повышающие при $i < 1$ и $n_1 < n_2$, или **мультипликаторы**. В большинстве случаев частоту вращения требуется понизить, поэтому редукторы используются значительно чаще, чем мультипликаторы.

Кроме соотношений (2.1) и (2.2), в расчете передач часто используют следующие зависимости:

$$T = P/\omega; \quad (2.3)$$

$$\omega = \pi n/30; \quad (2.4)$$

$$T_2 = T_1 i \eta, \quad (2.5)$$

где T – крутящий момент на валу передачи.

Формула (2.5) выражает связь между крутящими моментами на выходном валу T_2 и на входном валу T_1 .

Механические передачи делятся на **передачи трением** (фрикционные, ременные) и **передачи зацеплением** (зубчатые, червячные, цепные, винтовые). Передачи зацеплением по сравнению с передачами трением обладают повышенной нагрузочной способностью (или меньшими размерами при равной мощности), обеспечивают высокую точность и большую величину передаточного отношения, могут использоваться в широком диапазоне скоростей.

К недостаткам их можно отнести сложность изготовления, шум при высоких скоростях, неспособность компенсировать динамические нагрузки (жесткость).

Достоинствами передач зацеплением обуславливается их преимущественное использование в ММ и роботах. Поэтому далее о передачах трением дано только общее представление, а передачи зацеплением рассмотрены подробно.

2.2. Ременные передачи

Ременная передача (рис. 2.1) состоит из двух **шкивов** – ведущего 1 и ведомого 2, закрепленных на валах, и **ремня** 3, охватывающего шкивы.

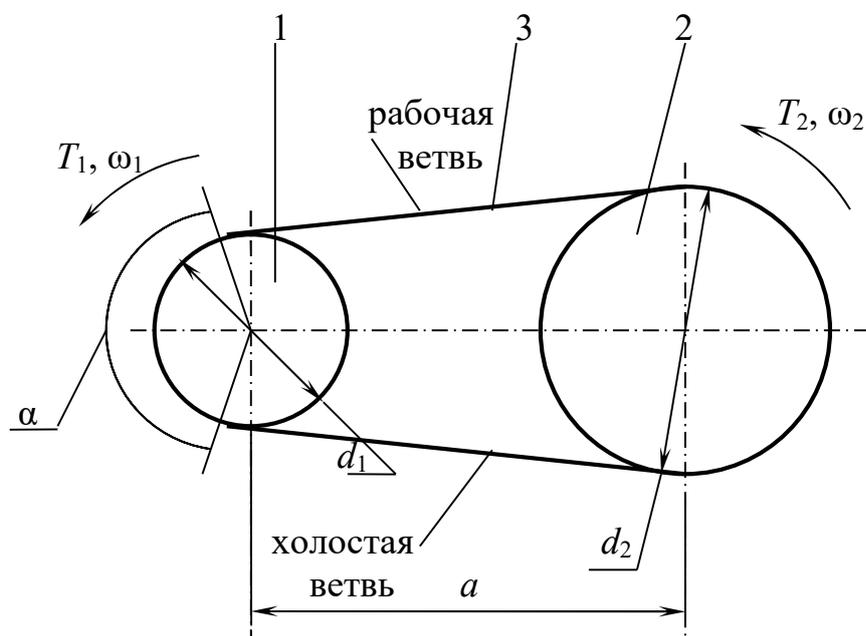


Рис. 2.1. Схема ременной передачи

Нагрузка передается силами трения, возникающими между ремнем и шкивами вследствие натяжения ремня. Передача также может включать в себя устройство для обеспечения требуемой силы натяжения ремня (**натяжное устройство**).

По форме поперечного сечения ремня различают передачи **плоскоременные** (рис. 2.2, *а*), **клиноременные** (рис. 2.2, *б*), **поликлиноременные** (рис. 2.2, *в*), **круглоременные**, а также передачи **пленочными ремнями**. Преимущественное распространение имеют передачи первыми двумя видами ремней.

Клиноременная передача по сравнению с плоскоременной имеет важные преимущества:

- бóльшие силы трения ремня по шкиву при равных силах натяжения, а следовательно, передача бóльших крутящих моментов и мощностей;
- передача осуществляется, как правило, несколькими клиновыми ремнями, в результате чего повышается ее надежность (выход из строя одного ремня еще не означает остановку механизма, а тем более, аварию).

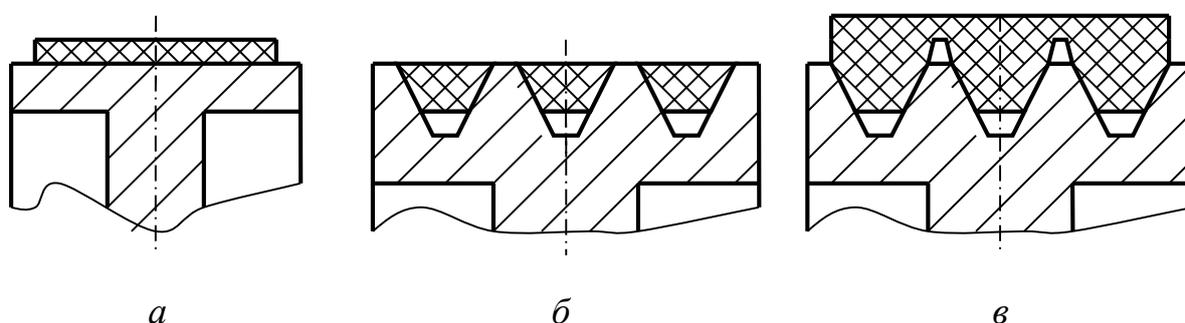


Рис. 2.2. Формы поперечного сечения ремней

К преимуществам плоскоременной следует отнести:

- возможность обеспечения значительных межосевых расстояний (размер *a* на рис. 2.1);
- возможность создания передач с непараллельными осями шкивов.

По способу натяжения ремней различают передачи **с натяжением при сборке**, **с периодическим подтягиванием** и **с автоматическим поддержанием натяжения**.

Способ натяжения ремня при сборке передачи самый простой: ремень с усилием надевают на шкивы и подтягивание его по мере износа и неупругой вытяжки не предусматривают. Нагрузочная способность такой передачи понижается, т.к. со временем натяжение ремня ослабевает.

Более совершенной в конструктивном отношении является передача, в которой возможно периодическое подтягивание ремня. Как правило, это достигается перемещением одного из шкивов, чаще – ведущего, с последующим закреплением на новом месте.

Передача с автоматическим поддержанием необходимого натяжения обычно содержит устройство в виде натяжного (плоскоременная передача) или оттяжного (клиноременная передача) ролика, воздействующего на холостую ветвь ремня. Поджатие ролика к ремню обеспечивается пружиной или грузом.

Установка оттяжного ролика отрицательно сказывается на долговечности ремней, поэтому в клиноременных передачах чаще используют периодическое подтягивание.

К основным геометрическим параметрам ременной передачи относятся **межосевое расстояние a , диаметры шкивов d_1 и d_2 , длина ремня L , угол обхвата ремнем малого шкива α .**

Рекомендуемые величины a :
для плоскоремennых передач

$$a \geq 2(d_1 + d_2); \quad (2.6)$$

для клиноременных передач

$$0,55(d_1 + d_2) + h \leq a \leq 2(d_1 + d_2), \quad (2.7)$$

где h – высота сечения ремня.

Точное значение передаточного отношения ременной передачи

$$i = \frac{n_1}{n_2} = \frac{d_2}{d_1(1 - \varepsilon)}, \quad (2.8)$$

где ε – коэффициент упругого скольжения ремня по шкиву.

При нормальной работе передачи $\varepsilon = 0,01 \dots 0,03$.

Соотношение крутящих моментов на шкивах

$$T_1 = \frac{T_2}{i\eta}, \quad (2.9)$$

где η – КПД передачи.

Плоскоремennые передачи имеют $\eta \approx 0,97$, клиноременные – $\eta \approx 0,96$.

Методики расчета ременных передач изложены в [1, 4].

2.3. Цилиндрические зубчатые передачи

2.3.1. Геометрические параметры цилиндрических зубчатых передач

Цилиндрические зубчатые передачи внешнего и внутреннего зацепления относятся к преобразователям вращательного движения с **параллельными осями колес**. Частным случаем такой передачи считают **реечное зацепление**, в котором одно из колес имеет бесконечно большой радиус, вследствие чего вырождается в прямолинейную **зубчатую рейку**.

По форме профиля зуба различают передачи **эвольвентные, циклоидные** и **Новикова**. В зацеплениях эвольвентном и циклоидном боковые стороны

профиля зуба очерчены соответственно по эвольвенте и циклоиде. В зацеплении Новикова профиль зуба образован дугами окружностей.

Циклоидное зацепление применяется в кинематических передачах приборов. Наибольшее распространение получило эвольвентное зацепление: оно позволяет создавать достаточно прочные и малогабаритные преобразователи движения и обладает существенными технологическими преимуществами. Наиболее высокие прочностные характеристики имеет зацепление Новикова, однако оно значительно сложнее в изготовлении. Далее будут рассматриваться передачи с эвольвентными зубьями.

По расположению зубьев на колесах различают цилиндрические передачи **прямозубые** (зуб расположен по образующей цилиндра), **косозубые** (зуб расположен по винтовой линии) и **шевронные** (рис. 2.3).

Меньшее зубчатое колесо пары (рис. 2.4) называется **шестерней**, большее – **зубчатым колесом** (или просто **колесом**). Параметрам шестерни присваивается индекс 1, параметрам колеса – индекс 2.

Термины, определения и методы расчета геометрических параметров зубчатых передач стандартизованы.

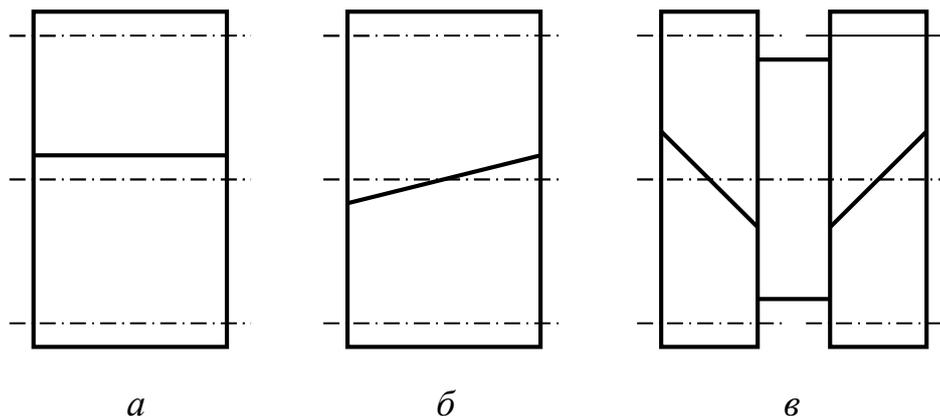


Рис. 2.3. Разновидности цилиндрических зубчатых колес по расположению зубьев:
a – прямозубое; *б* – косозубое; *в* – шевронное

Рассмотрим сначала прямозубую передачу, а затем – особенности геометрии косозубой передачи.

Числа зубьев – z_1 и z_2 .

Передаточное отношение от шестерни к колесу, называемое **передаточным числом**, равно

$$u = \frac{z_2}{z_1} . \quad (2.10)$$

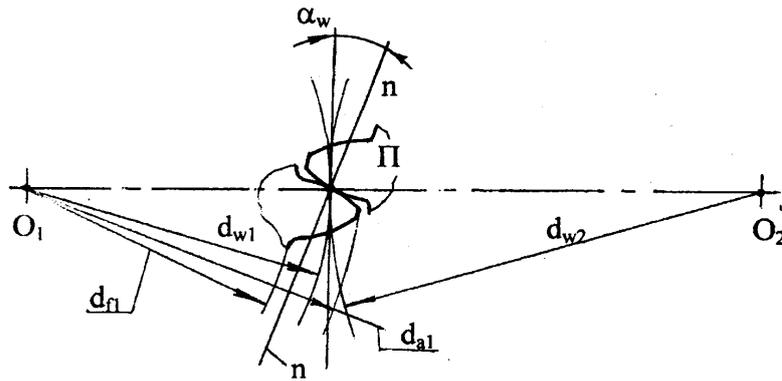


Рис. 2.4. Основные геометрические параметры
прямозубой передачи

Делительный окружной шаг зубьев p есть расстояние между сходственными точками двух соседних зубьев по дуге делительной окружности.

Делительной окружностью называется окружность, по которой производится деление заготовки на зубья. По дуге делительной окружности толщина зуба равна толщине впадины (и равна $\frac{p}{2}$).

Основной характеристикой размеров зубьев является **модуль**

$$m = \frac{p}{\pi}. \quad (2.11)$$

Значения модулей указаны в ГОСТ 9563.

Для шестерен и колес без смещения исходного контура (о смещении исходного контура см. ниже):

высота зуба

$$h = 2,25m. \quad (2.12)$$

В прямозубой передаче диаметры делительных окружностей

$$\begin{aligned} d_1 &= mz_1, \\ d_2 &= mz_2; \end{aligned} \quad (2.13)$$

диаметры окружностей вершин зубьев

$$\begin{aligned} d_{a1} &= m(z_1 + 2) = d_1 + 2m, \\ d_{a2} &= m(z_2 + 2) = d_2 + 2m; \end{aligned} \quad (2.14)$$

диаметры **окружностей впадин**

$$\begin{aligned}d_{f1} &= m(z_1 - 2,5) = d_1 - 2,5m, \\d_{f2} &= m(z_2 - 2,5) = d_2 - 2,5m;\end{aligned}\tag{2.15}$$

межосевое расстояние

$$a_w = 0,5m(z_1 + z_2).\tag{2.16}$$

Кроме того, выделяют **начальные окружности**, по которым шестерня и колесо обкатываются в процессе вращения. Диаметры начальных окружностей равны

$$\begin{aligned}d_{w1} &= \frac{2a_w}{u + 1}, \\d_{w2} &= 2a_w - d_{w1}.\end{aligned}\tag{2.17}$$

Для передач без смещения $d_1 = d_{w1}$; $d_2 = d_{w2}$.

Точка касания начальных окружностей, обозначенная буквой П на рис. 2.4, называется **полюсом зацепления**.

Общая нормаль $n-n$ к контактирующим поверхностям зубьев, проведенная через точку П, называется **линией зацепления**. При вращении шестерни и колеса точки контакта зубьев находятся на линии $n-n$. Угол между линией зацепления и перпендикуляром к **линии центров** O_1-O_2 – **угол зацепления** α_w . Стандартная величина $\alpha_w = 20^\circ$.

На рис. 2.5 показано расположение двух соседних зубьев косозубого колеса. Сечения зубьев тремя плоскостями – нормальной $n-n$, торцовой $t-t$ и осевой $a-a$ – дают соответственно **нормальный модуль** m_n , **торцовый модуль** m_t и **осевой модуль** m_a . В нормальном сечении профиль косоугого зуба совпадает с профилем прямого зуба, поэтому m_n должен быть стандартным. В торцовом и осевом сечениях модули зависят от угла наклона зуба β , стандарт на них не распространяется.

Особое значение имеют геометрические параметры в торцовом сечении:
модуль торцовый

$$m_t = \frac{m_n}{\cos \beta};\tag{2.18}$$

диаметр делительный

$$d = m_t z = \frac{m_n z}{\cos \beta}; \quad (2.19)$$

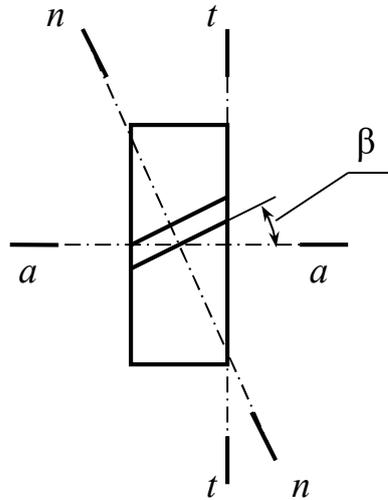


Рис. 2.5. Сечения зубьев косозубого колеса нормальной, торцовой и осевой плоскостями

шаг окружной

$$p_t = \frac{p_n}{\cos \beta}. \quad (2.20)$$

Геометрические параметры зубчатой передачи не исчерпываются указанными выше. Подробнее о геометрии зубчатых передач см., напр., [5].

Рассмотрим, как число зубьев влияет на их форму.

Одним из наиболее технологичных и широко применяемых способов изготовления зубчатых колес является так называемый **способ обкатки**. Суть способа сводится к тому, что зубонарезающий инструмент в виде зубчатой рейки или шестерни вводится «в зацепление» с заготовкой, и перемещения инструмента и заготовки в процессе обработки подобны перемещениям пары деталей, находящихся в зацеплении.

При изготовлении обкаткой боковые стороны профиля зуба получаются эвольвентными. С увеличением числа зубьев колеса боковые стороны приближаются к прямолинейным, и в предельном случае, когда $z = \infty$ (зубчатая рейка), профиль приобретает форму равнобокой трапеции. Наоборот, с уменьшением числа зубьев толщина зуба у основания и вершины уменьшается, кривизна профиля увеличивается. Когда z становится меньше некоторого минимального значения z_{\min} , зубья инструмента, проворачиваясь во впадине заготовки, удаляют материал из ножки зуба. Это явление называется **подрезанием ножки**, оно существенно снижает прочность зуба. Для прямозубых передач считают $z_{\min} = 17$.

При необходимости выполнения $z < z_{\min}$, а также в некоторых других случаях применяют смещение нарезающего инструмента: инструмент отодвигают от положения, соответствующего нарезанию без смещения, на расстояние xm , где x – коэффициент смещения исходного контура. Смещение считают положительным, если оно направлено от центра, и отрицательным, если к центру заготовки. Шестерни нарезают при положительном смещении, что позволяет существенно изменить форму зуба: он становится короче и толще, подрезание ножки устраняется.

Нарезание зубьев со смещением является определенным усложнением процесса изготовления, поэтому рекомендуется по возможности не назначать z меньше z_{\min} . Увеличения числа зубьев практически всегда можно достичь уменьшением модуля.

2.3.2. Силы, действующие в зубчатом зацеплении

На рис. 2.6 показано прямозубое зацепление.

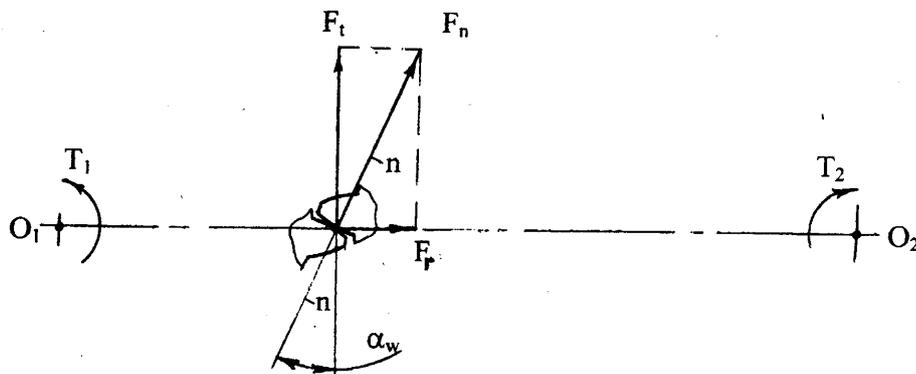


Рис. 2.6. Силы в прямозубом зацеплении

Силу нормального давления зуба шестерни на зуб колеса F_n можно разложить на две составляющих:
окружную силу

$$F_t = \frac{2T_2}{d_2}; \quad (2.21)$$

радиальную силу

$$F_r = F_t \operatorname{tg} \alpha_w. \quad (2.22)$$

Сама сила F_n равна

$$F_n = \frac{F_t}{\cos \alpha_w}. \quad (2.23)$$

В косозубом зацеплении (рис. 2.7) нормальную силу раскладывают на три составляющих:
 окружную силу – см. (2.21);
 радиальную силу

$$F_r = \frac{F_t \operatorname{tg} \alpha_w}{\cos \beta}; \quad (2.24)$$

осевую силу

$$F_a = F_t \operatorname{tg} \beta. \quad (2.25)$$

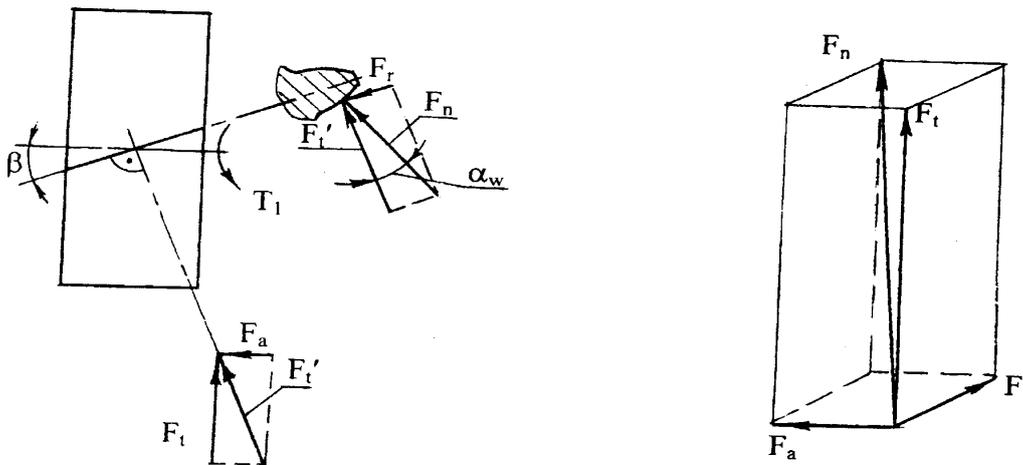


Рис. 2.7. Силы, действующие на зуб косозубого колеса

Нормальная сила является диагональю параллелепипеда с ребрами, равными F_t , F_r и F_a , и может быть найдена так:

$$F_n = \frac{F_t}{\cos \alpha_w \cos \beta}. \quad (2.26)$$

2.3.3. Виды разрушения зубьев

Основной вид разрушения поверхности зубьев при хорошей смазке передачи, надежно защищенной от попадания пыли и грязи, – **усталостное выкрашивание**. Зубья таких передач разделены тонким слоем масла, устраняющим металлический контакт. Износ зубьев мал. Передача работает длительное время до появления усталости в поверхностных слоях зубьев. На поверхности появляются небольшие углубления, которые растут и превращаются в раковины. Выкрашиванию способствует смазка, она запрессовывается зубьями в трещины и своим давлением отделяет частицы металла. Причина усталостного выкрашивания – контактные напряжения σ_H .

Основные меры предупреждения выкрашивания: расчет передачи на усталость по контактным напряжениям; применение материалов с повышенной твердостью поверхности; повышение точности изготовления и монтажа передач. Во многих случаях выкрашивания можно избежать, если предусмотреть приработку зубьев в процессе эксплуатации передачи (о приработке см. ниже).

Поломка зубьев связана с напряжениями изгиба σ_F . Различают **полный** (по всей длине) и **угловой** излом. Одна из причин полного излома – перегрузки ударного или статического характера. Другая причина – усталостная поломка от действия переменных напряжений. Причиной углового излома являются погрешности передачи, в результате которых нагрузка воспринимается не всей длиной зуба, а концентрируется на одном из его углов.

Поломку от перегрузок предупреждают защитой передачи посредством различных предохранительных устройств или учетом перегрузок при расчете; поломку от переменных напряжений предупреждают определением размеров из расчета на усталость. К общим мерам относятся: увеличение модуля, положительное смещение при нарезании зубьев, термообработка, устранение концентраторов напряжений (рисок от обработки, раковин в отливках, микротрещин от термообработки). Углового излома можно избежать повышением точности сборки передачи (в частности, обеспечением параллельности осей колес) и применением зубьев со срезанными углами.

Открытые передачи, а также закрытые, но недостаточно защищенные от попадания абразивных частиц, выходят из строя в основном из-за **абразивного износа**. По мере износа зубьев увеличиваются зазоры в зацеплении, появляется шум, возрастают динамические нагрузки. Толщина зубьев уменьшается, и соответственно снижается их прочность.

Основные меры борьбы с износом: повышение твердости поверхности зубьев, защита от загрязнения, применение специальных смазочных материалов. Большое значение имеет своевременное диагностирование сверхнормативного износа и замена изношенных колес.

Кроме перечисленных видов разрушения зубьев, наблюдаются такие, как заедание, пластический сдвиг, отслаивание твердого поверхностного слоя. Однако при грамотном расчете, качественном изготовлении и правильной эксплуатации передачи вероятность этих разрушений значительно ниже.

2.3.4. Материалы зубчатых передач

В настоящее время установлено, что контактная прочность зубьев определяется в основном твердостью материала. Наибольшая твердость, а следовательно, наименьшие габариты и массу передачи можно получить при изготовлении колес из сталей, подвергнутых термической обработке.

В зависимости от твердости стальные зубчатые колеса разделяют на две основные группы: твердостью $HV < 350$ и твердостью $HV > 350$.

Твердость $HV < 350$ позволяет производить чистовое нарезание зубьев после термообработки, в результате чего можно получать высокую точность без

дорогих отделочных операций (шлифовки, притирки и т.п.). Колеса этой группы хорошо прирабатываются и не подвержены хрупкому разрушению при динамических нагрузках. Под **приработкой** понимают износ поверхностей зубьев передачи, приводящий к более равномерному распределению нагрузки по длине зуба, а потому существенно повышающий надежность зубчатых колес. Приработку зубьев широко используют в условиях индивидуального и мелкосерийного производства в мало- и средненагруженных передачах. Обычно для лучшей приработки твердость шестерни назначают на 20...50HB больше, чем твердость колеса.

Твердость материалов второй группы ($HB > 350$) обычно выражают в единицах HRC . Соотношение единиц HB и HRC таково: $1HRC \approx 10HB$. Специальными видами термообработки могут быть получены твердости 50...60 HRC , причем допускаемые контактные напряжения возрастают примерно в два раза, а нагрузочная способность передачи – в четыре раза по сравнению с передачей, изготовленной из материалов первой группы. Очевидно, что применение высокотвердых материалов является большим резервом повышения нагрузочной способности зубчатых передач.

К недостаткам материалов этой группы следует отнести плохую прирабатываемость и, как следствие, необходимость повышенной точности изготовления деталей передач и их монтажа. Кроме того, некоторые виды термообработки (объемная закалка, цементация) сопровождаются значительным короблением зубьев. Для исправления формы зубьев требуются дополнительные операции.

Данные по механическим характеристикам некоторых наиболее широко используемых сталей приведены в табл. 2.1. Материалы группы $HB < 350$ представлены нормализованными и улучшенными сталями, а группы $HB > 350$ – закаленными объемной или поверхностной закалкой, а также азотированными. Подробнее о режимах термообработки см., например, [6].

В зависимости от способа получения заготовки различают литые, кованные, штампованные колеса и колеса из круглого проката.

2.3.5. Методика расчета закрытой зубчатой передачи

2.3.5.1. Общие положения

Настоящая методика основана на ГОСТ 21354 и ГОСТ 2185, предназначена для расчета на усталостную и статическую прочность эвольвентных цилиндрических зубчатых передач и может быть использована студентами при конструировании ММ в ходе курсового и дипломного проектирования.

Большую часть общего количества выпускаемых в настоящее время зубчатых передач составляют одно- и двухступенчатые редукторы, поэтому в методике рассматривается двухступенчатый редуктор.

Рекомендуется следующий порядок расчета:

– выбор двигателя по требуемой номинальной мощности, заданной частоте вращения и условиям работы;

- кинематический расчет редуктора, разбивка его передаточного числа по ступеням;
- выбор материалов для шестерен и колес, определение допускаемых напряжений;
- определение крутящих моментов на шестернях и колесах ступеней;
- проектировочный расчет передачи тихоходной ступени;
- проверочные расчеты передачи тихоходной ступени;
- проектировочный расчет передачи быстроходной ступени;
- проверочные расчеты передачи быстроходной ступени.

Таблица 2.1

Механические характеристики сталей

Группа сталей	Марка стали	Термообработка	Твердость		σ_B , МПа	σ_T , МПа
			поверхности	сердцевины		
<i>HB < 350</i>	35	Н	163...192HB		550	270
	40	У	192...228HB		700	400
	45	Н	179...207HB		600	320
		У	235...262HB		780	540
		У	269...302HB		890	650
	40X	У	235...262HB		790	640
		У	269...302HB		900	750
	40XH	У	235...262HB		800	650
		У	269...302HB		920	750
	35XM	У	235...262HB		800	670
		У	269...302HB		920	790
	35Л	Н	163...207HB		550	270
	40Л	Н	147HB		520	295
	45Л	У	207...235HB		680	440
40ГЛ	У	235...262HB		850	600	
<i>HB > 350</i>	40X	У + ТВЧ	45...50HRC	269...302HB	900	750
		А	50...59HRC	269...302HB	1000	800
	40XH	З	48...54HRC		1600	1400
		У + ТВЧ	48...54HRC	269...302HB	920	750
	35XM	З	45...53HRC		1600	1400
		У + ТВЧ	48...54HRC	269...302HB	920	790
	38XMЮА	З	45...53HRC		1700...1950	1350...1600
		А	57...67HRC	30...35HRC	1050	900
Обозначение термообработки: У – улучшение; Н – нормализация; З – закалка объемная; ТВЧ – закалка с нагревом токами высокой частоты; А – азотирование						

2.3.5.2. Исходные данные для расчета

Задание на курсовой проект содержит следующие **обязательные** данные, используемые как исходные при расчете редуктора:

- кинематическая схема ММ, дающая возможно более полное представление о том, как передается вращение на ведущий вал и снимается с ведомого вала передачи;
- номинальный крутящий момент на выходном валу T_T , Нм;
- частота вращения выходного вала n_T , об/мин;
- срок службы ММ $T_{сл}$, лет;
- нагрузочная диаграмма (циклограмма), отражающая изменение крутящего момента на выходном валу в течение рабочей смены (рабочего цикла);
- максимальные допустимые значения кинематической погрешности $F'_{io\max}$ и мертвого хода $J_{t\max}$, МКМ.

Кроме указанных, задание может включать в себя **дополнительные требования**, например, кратность максимального допустимого пикового момента, направление вращения выходного вала, максимальные допустимые габаритные размеры ММ и т.д.

2.3.5.3. Выбор двигателя

Тип двигателя ММ проектант выбирает самостоятельно, руководствуясь рекомендациями учебно-методической литературы [4, 5]. В настоящей методике для определенности принят наиболее широко распространенный в машиностроении трехфазный асинхронный короткозамкнутый электродвигатель.

Требуемую номинальную мощность двигателя вычисляют по формуле

$$P_{тр} = \frac{T_T n_T}{\eta_6 \eta_T \eta_{п}^3}, \quad (2.27)$$

где η_6 , η_T – КПД быстроходной и тихоходной ступеней соответственно; $\eta_{п} = 0,99$ – КПД пары подшипников качения.

Как правило, в ММ применяются зубчатые передачи степеней точности не ниже 7, для которых $\eta_6 = \eta_T = 0,98 \dots 0,99$.

Затем принимают типоразмер двигателя по условию

$$P_{дв} \geq P_{тр}, \quad (2.28)$$

где $P_{дв}$ – номинальная мощность двигателя по каталогу.

Допустимо превышение требуемой мощности над номинальной, если выполняется условие

$$\frac{P_{\text{тр}} - P_{\text{дв}}}{P_{\text{дв}}} \times 100\% \leq [\Delta P], \quad (2.29)$$

где $[\Delta P]$ – допустимая перегрузка двигателя принятого типа.

2.3.5.4. Кинематический расчет редуктора Расчетное передаточное число редуктора

$$u_p = \frac{n_{\text{дв}}}{n_{\text{т}}}, \quad (2.30)$$

где $n_{\text{дв}}$ – частота вращения вала двигателя.

В том случае, если в исходных данных отсутствует требование абсолютно точного обеспечения заданной величины $n_{\text{т}}$, следует для дальнейшего проектирования принимать передаточное число редуктора u в соответствии со стандартным рядом передаточных чисел по условию

$$\frac{|u - u_{\text{ст}}|}{u_{\text{ст}}} \cdot 100\% \leq \Delta u, \quad (2.31)$$

где $u_{\text{ст}}$ – ближайшее к u_p значение передаточного числа из стандартного ряда; Δu – допустимое отклонение передаточного числа от стандартного значения.

Для цилиндрических зубчатых передач при $u_p \leq 4,5$ отклонение $\Delta u = 2,5\%$, при $u_p > 4,5$ $\Delta u = 4\%$.

В том случае, если условие (2.31) выполняется при подстановке вместо u значения u_p , можно принять либо $u = u_p$, либо $u = u_{\text{ст}}$. И то, и другое решение будет правомерно.

Для принятого u далее следует, пользуясь рекомендациями табл. 2.2, 2.3, определить передаточные числа ступеней: быстроходной $u_{\text{б}}$ и тихоходной $u_{\text{т}}$.

Таблица 2.2
Передаточные числа ступеней косозубых трехосных редукторов

$$\left(\frac{a_{\text{вт}}}{a_{\text{вб}}} = 1,4 \right)$$

Передаточное число ступени	Передаточное число редуктора u										
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5
$u_{\text{б}}$	3,55	4		4,5	5	5,6		6,3	7,1		8
$u_{\text{т}}$	2,8		3,15			3,55			4		

Таблица 2.3

Передаточные числа ступеней косозубых соосных редукторов

$$\left(\begin{array}{l} \Psi_{ba\Gamma} = 2 \\ \Psi_{ba\delta} \end{array} \right)$$

Передаточное число ступени	Передаточное число редуктора u												
	10	11,2	12,5	14	16	18	20	22,4	25	28	31,5		
u_{δ}	4,5		5		5,6		6,3		7,1		8		9
u_{Γ}	2,24		2,5				2,8				3,15		3,55

Далее следует найти частоты вращения и угловые скорости валов:
быстроходного

$$\begin{aligned} n_{\delta} &= n_{дв}, \\ \omega_{\delta} &= \frac{\pi n_{\delta}}{30}; \end{aligned} \quad (2.32)$$

промежуточного

$$n_{\Pi} = \frac{n_{\delta}}{u_{\delta}}, \quad (2.33)$$

$$\omega_{\Pi} = \frac{\omega_{\delta}}{u_{\delta}}; \quad (2.34)$$

тихоходного

$$n_{\Gamma} = \frac{n_{\Pi}}{u_{\Gamma}}, \quad (2.35)$$

$$\omega_{\Gamma} = \frac{\omega_{\Pi}}{u_{\Gamma}}. \quad (2.36)$$

2.3.5.5. Материалы шестерен и колес. Допускаемые напряжения

Марки сталей и режимы термообработки для шестерен и колес редуктора назначают по рекомендациям п. 2.3.4 и данным табл. 2.1.

Допускаемое контактное напряжение при расчете на выносливость определяют по формуле

$$[\sigma_H] = \frac{\sigma_{H\lim b}}{S_H} K_{HL}, \quad (2.37)$$

где σ_{Hlimb} – базовый предел контактной выносливости поверхности зубьев, соответствующий базовому числу циклов перемены напряжений N_{H0} ; S_H – коэффициент безопасности; K_{HL} – коэффициент долговечности.

Значения N_{H0} определяют по табл. 2.4, σ_{Hlimb} – по табл. 2.5.

Коэффициент $S_H = 1,1$ для нормализованных, улучшенных и объемно закаленных сталей (для материалов с однородной структурой); $S_H = 1,2$ для закаленных с нагревом ТВЧ и азотированных сталей (для материалов с неоднородной структурой).

Таблица 2.4

Значения N_{H0} , млн. циклов

Твердость зубьев	<i>HB</i>	200	250	300	–	–	–	–	–	–
	<i>HRC</i>	–	–	–	36	42	47	52	56	59
N_{H0}		10	12,5	25	35	50	65	85	110	150

Таблица 2.5

Значения σ_{Hlimb}

Вид термообработки	Твердость поверхности зубьев	Группа сталей	σ_{Hlimb} , МПа
Нормализация, улучшение	$HB < 350$	Углеродистые и легированные	$2HB_{cp} + 70$
Закалка объемная	$38...50HRC$		$18HRC_{cp} + 150$
Закалка с нагревом ТВЧ	$40...56HRC$		$17HRC_{cp} + 200$
Азотирование	$57...67HRC$	Легированные	$16HRC_{cp}$
<i>HB_{cp}, HRC_{cp}</i> – средние значения твердости в диапазоне (см. табл. 2.1)			

Коэффициент долговечности определяют из выражения

$$1 \leq K_{HL} = \sqrt[6]{\frac{N_{H0}}{N_{HE}}} \leq K_{HLmax}, \quad (2.38)$$

где N_{HE} – эквивалентное число циклов перемены напряжений; K_{HLmax} – максимальное допустимое значение коэффициента долговечности, зависящее от вида термообработки (при объемном упрочнении $K_{HLmax} = 2,6$; при поверхностном упрочнении $K_{HLmax} = 1,8$).

Как видно из (2.38), K_{HLmax} не может быть меньше единицы, поэтому при $N_{H0} < N_{HE}$ считают $K_{HL} = 1$.

Величина N_{HE} зависит от нагрузочной диаграммы. При постоянной нагрузке

$$N_{HE} = 60nct, \quad (2.39)$$

где n – частота вращения колеса (шестерни), $[\sigma_H]$ которого определяется, об/мин; c – число зацеплений зуба за один оборот колеса (шестерни); t – заданный срок службы редуктора, ч.

В том случае, если задана ступенчатая нагрузочная диаграмма, N_{HE} определяют так:

$$N_{HE} = 60c \sum \left(\frac{T_i}{T_1} \right)^3 n_i t_i, \quad (2.40)$$

где T_i – крутящий момент на i -ой ступени циклограммы; T_1 – наибольший момент на циклограмме, учитываемый в расчете на выносливость; n_i , t_i – соответствующие моменту T_i частота и время работы.

Для **прямозубой** ступени, а также для **косозубой с небольшой разностью твердости** зубьев шестерни и колеса в качестве расчетного принимают **меньшее** из двух допускаемых напряжений, определенных по материалу шестерни $[\sigma_H]_1$ и колеса $[\sigma_H]_2$.

Для **косозубой** ступени с **большой разностью твердости** зубьев шестерни и колеса в качестве расчетного принимают напряжение, определенное по (2.41):

$$[\sigma_H]_{\min} \leq 0,45([\sigma_H]_1 + [\sigma_H]_2) \leq 1,25[\sigma_H]_{\min}, \quad (2.41)$$

где $[\sigma_H]_{\min}$ – меньшее из значений $[\sigma_H]_1$ и $[\sigma_H]_2$.

Допускаемое напряжение изгиба при расчете на выносливость определяют по формуле

$$[\sigma_F] = \frac{\sigma_{F\limb}}{S_F} K_{FC} K_{FL}, \quad (2.42)$$

где $\sigma_{F\limb}$ – базовый предел выносливости зубьев по излому от напряжений изгиба (см. табл. 2.6); S_F – коэффициент безопасности; K_{FC} – коэффициент влияния двухстороннего приложения нагрузки; K_{FL} – коэффициент долговечности.

Коэффициент S_F принимают в зависимости от вида термообработки из диапазона 1,7...2,2 (верхнее значение – для литых колес).

Коэффициент $K_{FC} = 1$ для односторонней нагрузки, $K_{FC} = 0,7...0,8$ для реверсивной нагрузки (большие значения при $HB > 350$).

Коэффициент K_{FL} определяют по (2.43):

$$1 \leq K_{FL} = \sqrt[m]{\frac{4 \cdot 10^6}{N_{FE}}} \leq K_{FL\max}, \quad (2.43)$$

где m – показатель степени; N_{FE} – эквивалентное число циклов нагружения напряжениями изгиба; K_{FLmax} – максимальное допустимое значение коэффициента долговечности.

При $HB \leq 350$, а также для шестерен и колес со шлифованными зубьями $m = 6$, $K_{FLmax} = 2,0$; при $HB > 350$, а также для шестерен и колес с нешлифованными зубьями $m = 9$, $K_{FLmax} = 1,6$.

При постоянной нагрузке значение N_{FE} находят по формуле (2.39), при изменении нагрузки по нагрузочной диаграмме – по формуле (2.44):

$$N_{FE} = 60c \sum \left(\frac{T_i}{T_1} \right)^m n_i t_i. \quad (2.44)$$

Предельное допускаемое контактное напряжение для проверки ступени на прочность при перегрузках (пиковых нагрузках) равно:

Таблица 2.6

Значения σ_{Flimb}

Вид термообработки	Твердость зубьев		Группа сталей	σ_{Flimb} , МПа
	поверхность	сердцевина		
Нормализация, улучшение	180...350HB		Углеродистые и легированные	1,8HB _{ср}
Закалка объемная	45...55HRC			550...600
Закалка с нагревом ТВЧ	40...56HRC	HB ≤ 350		1,8HB _{ср}
Азотирование	57...67HRC	24...40HRC	Легированные	43...49HRC _{ср}
<i>HB_{ср}, HRC_{ср} – средние значения твердости в диапазоне (см. табл. 2.1)</i>				

для нормализованных, улучшенных или объемно закаленных зубьев

$$[\sigma_{Hmax}] = 2,8\sigma_T; \quad (2.45)$$

для зубьев, закаленных с нагревом ТВЧ

$$[\sigma_{Hmax}] = 44HRC_{ср}; \quad (2.46)$$

для азотированных зубьев

$$[\sigma_{Hmax}] = 35HRC_{ср}. \quad (2.47)$$

Предельное допускаемое напряжение изгиба для проверки ступени на прочность при перегрузках определяют следующим образом:

$$[\sigma_{F \max}] = \frac{\sigma_{F \lim b}}{S_{ST}} Y_{N \max} k_{ST}, \quad (2.48)$$

где $Y_{N \max}$ – максимальное возможное значение коэффициента долговечности (для объемной термообработки $Y_{N \max} = 4,0$; для поверхностной термообработки $Y_{N \max} = 2,5$); k_{ST} – коэффициент влияния частоты приложения пиковой нагрузки (при многократном – порядка 1000 – действии перегрузок $k_{ST} = 1$); S_{ST} – коэффициент запаса прочности (обычно принимают $S_{ST} = 1,75$).

В настоящее время в преобразователях движения ММ и роботов применяются практически исключительно косозубые передачи, поэтому именно они и рассматриваются далее в методике.

2.3.5.6. Крутящие моменты на шестернях и колесах ступеней

Крутящий момент на колесе тихоходной ступени равен

$$T_{2T} = \frac{T_T}{\eta_{\pi}}; \quad (2.49)$$

крутящий момент на шестерне тихоходной ступени и колесе быстроходной ступени

$$T_{1T} = T_{26} = \frac{T_{2T}}{u_T \eta_T \eta_{\pi}}; \quad (2.50)$$

крутящий момент на шестерне быстроходной ступени

$$T_{16} = \frac{T_{26}}{u_6 u_T \eta_6 \eta_T \eta_{\pi}}. \quad (2.51)$$

2.3.5.7. Проектировочный расчет тихоходной ступени

Расчетное межосевое расстояние определяют по формуле

$$a'_{\text{вт}} = 430(u_T \pm 1)^3 \sqrt{\frac{T_{2T} K_{H\beta T}}{u_T^2 \psi_{\text{bat}} [\sigma_H]_T^2}}, \text{ мм}, \quad (2.52)$$

где $K_{H\beta T}$ – коэффициент неравномерности распределения нагрузки по длине зуба колеса тихоходной ступени (табл. 2.8); ψ_{bat} – коэффициент ширины колеса тихоходной ступени относительно ее межосевого расстояния; $[\sigma_H]_T$ – допускаемое контактное напряжение для тихоходной ступени.

Знак плюс в (2.52) соответствует внешнему зацеплению, минус – внутреннему зацеплению.

Смысл коэффициента ψ_{bat} проясняет формула

$$\psi_{bat} = \frac{b_{2T}}{a_{WT}}, \quad (2.53)$$

где b_{2T} – ширина венца колеса тихоходной ступени.

Стандартный ряд значений ψ_{bat} : 0,100; 0,125; 0,160; 0,200; 0,250; 0,315; 0,400; 0,500; 0,630; 0,800; 1,00; 1,25. Следует применять в расчете коэффициент ψ_{ba} из этого ряда в соответствии с рекомендациями табл. 2.7.

Далее следует принять **стандартное межосевое расстояние** a_{WT} , ближайшее к расчетному a'_{WT} , по табл. 2.9.

Определение **основных геометрических параметров** ступени целесообразно начать с выбора нормального модуля по рекомендации

$$1,50 \leq m_{WT} = (0,01 \dots 0,02) a_{WT}, \text{ мм}, \quad (2.54)$$

из стандартного ряда: 1,50; 1,75; 2,00; 2,25; 2,50; 2,75; 3,00; 3,50; 4,00; 4,50; 5,00; 5,50; 6,00; 7,00; 8,00...

Таблица 2.7

Рекомендуемые значения ψ_{ba}

Расположение колеса относительно опор	Твердость рабочих поверхностей зубьев	
	$HB_2 \leq 350$	$HB_2 > 350$
Симметричное	0,200...0,400	0,200...0,315
Несимметричное		0,200; 0,250
Консольное	0,200; 0,250	0,160; 0,200
Примечание: в шевронных передачах ψ_{ba} следует увеличить в 1,1...1,4 раза; для подвижных колес коробок передач $\psi_{ba} = 0,100 \dots 0,200$		

В обоснованных случаях для передач 6 и 7 степеней точности могут быть приняты модули по ГОСТ 9563 менее 1,5 мм.

Затем нужно предварительно задать значение угла наклона зубьев $\beta' = 8 \dots 15^\circ$ (обычно принимают $\beta' = 10^\circ$), после чего определить **расчетное суммарное число зубьев** передачи по формуле (2.55):

$$z'_{ст} = \frac{2a_{WT} \cos \beta'_T}{m_{WT}}. \quad (2.55)$$

Полученное число $z'_{ст}$ округлить до ближайшего целого $z_{ст}$.

Уточненную величину угла наклона зубьев определяют по (2.56)

$$\beta_T = \arccos \frac{z_{ст} m_{нТ}}{2a_{вт}} \quad (2.56)$$

с точностью до угловых секунд.

Таблица 2.8

Значения $K_{H\beta}$

Относительная ширина колеса ψ_{bd}	Шестерня расположена симметрично относительно опор		Шестерня расположена несимметрично относительно опор				Консольное расположение шестерни или колеса	
			весьма жесткий вал		менее жесткий вал			
	твердость поверхностей зубьев HB_2		твердость поверхностей зубьев HB_2		твердость поверхностей зубьев HB_2		твердость поверхностей зубьев HB_2	
	> 350	< 350	> 350	< 350	> 350	< 350	> 350	< 350
0,2	1,00		1,01	1,00	1,06	1,02	1,15	1,07
0,4	1,01	1,00	1,05	1,02	1,12	1,05	1,35	1,15
0,6	1,03	1,01	1,09	1,04	1,20	1,08	1,60	1,24
0,8	1,06	1,03	1,14	1,06	1,27	1,12	1,85	1,30
1,0	1,10	1,04	1,18	1,08	1,37	1,15	—	
1,2	1,13	1,05	1,25	1,10	1,50	1,18		
1,4	1,15	1,07	1,32	1,13	1,60	1,23		
1,6	1,20	1,08	1,40	1,16	—	1,28		
Относительная ширина колеса $\psi_{bd} = 0,5\psi_{ba}(u + 1)$								

Таблица 2.9

Межосевые расстояния двухступенчатых трехосных цилиндрических редукторов

$a_{вб}, \text{ мм}$	50	63	80	100	112	125	140	160	180
$a_{вт}, \text{ мм}$	63	80	112	140	160	180	200	224	250
$a_{вб}, \text{ мм}$	200	224	250	280	315	355	400	450	500
$a_{вт}, \text{ мм}$	280	315	355	400	450	500	560	630	710

Расчетное число зубьев шестерни равно

$$z'_{1Т} = \frac{z_{ст}}{u_T + 1}. \quad (2.57)$$

Полученное число $z'_{1Т}$ округлить до ближайшего целого $z_{1Т}$.

Во избежание подрезания зубьев должно выполняться условие

$$z_{1Т} \geq 17 \cos^3(\beta_T). \quad (2.58)$$

Если (2.58) не выполняется, следует принять меньшее значение $m_{нт}$ из стандартного ряда и заново выполнить (2.55), (2.56) и (2.57).

Далее находят число зубьев колеса

$$z_{2г} = z_{ст} - z_{1г} \quad (2.59)$$

и **фактическое передаточное число** ступени

$$u_{фг} = \frac{z_{2г}}{z_{1г}},$$

которое проверяют по условию (2.31).

По (2.19) вычисляют $d_{1г}$, $d_{2г}$, после чего находят диаметры вершин зубьев

$$d_{a1г} = d_{1г} + 2m_{нт}; \quad (2.60)$$

$$d_{a2г} = d_{2г} + 2m_{нт}. \quad (2.61)$$

Ширину венца колеса предварительно определяют по выражению

$$b_{2г} = \psi_{bat} a_{вт} \quad (2.62)$$

и окончательно принимают ближайшее значение из ряда R_a40 номинальных линейных размеров ГОСТ 6636.

Ширину венца шестерни $b_{1г}$ принимают равной следующему за $b_{2г}$ размеру по указанному выше ряду.

2.3.5.8. Проверка тихоходной ступени на выносливость по контактным напряжениям

Действительное контактное напряжение в проектируемой передаче равно

$$\sigma_H = \frac{6160 Z_H Z_\varepsilon}{a_{вт}} \sqrt{\frac{T_{п} (u_{фг} \pm 1)^3}{u_{фг} b_{2г}} K_{H\alpha} K_{H\beta} K_{H\nu}}, \text{ МПа}, \quad (2.63)$$

где $Z_H = 1,77 \cos \beta$ – коэффициент формы сопряженных поверхностей зубьев; Z_ε – коэффициент суммарной длины контактных линий; $K_{H\alpha}$ – коэффициент распределения нагрузки между зубьями (табл. 2.10); $K_{H\nu}$ – коэффициент динамической нагрузки (табл. 2.11).

Для определения Z_ε необходимо найти коэффициент осевого перекрытия

$$\varepsilon_\beta = \frac{b_{2г} \sin \beta_\tau}{\pi m_{нт}} \quad (2.64)$$

и проследить, чтобы выполнялось условие $\varepsilon_\beta \geq 0,9$, в противном случае нагрузочная способность косозубой передачи резко снизится. Возможно, что для обеспечения выполнения этого условия в геометрию передачи придется вносить изменения.

Коэффициент Z_ε равен

$$Z_\varepsilon = \frac{1}{\sqrt{\varepsilon_\alpha}}, \quad (2.65)$$

где ε_α – коэффициент торцевого перекрытия, определяемый по формуле (2.66):

$$\varepsilon_\alpha = \left[1,88 - 3,2 \left(\frac{1}{z_1} \pm \frac{1}{z_2} \right) \right] \cos \beta. \quad (2.66)$$

Знак плюс в круглых скобках соответствует внешнему зацеплению, минус – внутреннему зацеплению.

Коэффициент $K_{H\alpha}$ принимают по табл. 2.10 для степени точности, которую назначают в зависимости от окружной скорости зуба

$$v_\pi = \frac{\omega_\pi d_{1T}}{2000}, \text{ м/с}, \quad (2.67)$$

по рекомендациям: $v_\pi \leq 4$ м/с – степень точности 9; $4 < v_\pi \leq 10$ м/с – степень точности 8; $10 < v_\pi \leq 15$ м/с – степень точности 7; $15 < v_\pi \leq 30$ м/с – степень точности 6.

Таблица 2.10

Значения $K_{H\alpha}$

Окружная скорость v , м/с	Степень точности			
	6	7	8	9
2,5	1,01	1,03	1,05	1,13
5,0	1,02	1,05	1,09	1,16
10	1,03	1,07	1,13	–
15	1,04	1,09	–	
20	1,05	1,12		
25	1,06	–		

Полученное значение контактного напряжения проверяют на соответствие условию

$$0,9[\sigma_H] \leq \sigma_H \leq 1,05[\sigma_H]. \quad (2.68)$$

В том случае, если σ_H не входит в указанные пределы, необходимо скорректировать параметры передачи. Обычно бывает достаточно изменить размеры b_{2T} и b_{1T} . Как правило, эти изменения невелики, и пересчета остальных размеров ступени не требуется.

Таблица 2.11

Значения K_{Hv}

Степень точности	Твердость HB_2	Окружная скорость зуба v_n , м/с					
		1	2	4	6	8	10
6	≤ 350	1,01	1,02	1,03	1,04	1,06	1,07
	> 350	1,00		1,02		1,03	1,04
7	≤ 350	1,01	1,02	1,04	1,06	1,07	1,08
	> 350	1,00	1,01	1,02	1,03		1,04
8	≤ 350	1,01	1,03	1,05	1,06	1,07	1,08
	> 350	1,01		1,02	1,03	1,04	1,05
9	≤ 350	1,01	1,03	1,05	–		
	> 350	1,01		1,02	–		

2.3.5.9. Проверка тихоходной ступени на выносливость по напряжениям изгиба

Проверку выполняют по «слабому» звену зубчатой передачи, у которого меньше отношение $\frac{[\sigma_F]}{Y_F}$, где Y_F – коэффициент формы зуба, определяемый для шестерни и колеса из табл. 2.12.

Таблица 2.12

Значения Y_F

$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F	$z/\cos^3\beta$	Y_F
17	4,26	21	4,01	28	3,82	40	3,70	80	3,61
18	4,20	22	4,00	30	3,80	45	3,68	100	3,60
19	4,11	24	3,92	32	3,78	50	3,65	150	
20	4,08	25	3,90	37	3,71	60	3,62	рейка	3,63

Действительное напряжение изгиба в зубе «слабого» звена проектируемой передачи равно

$$\sigma_F = 2000 Y_F Y_\beta Y_\varepsilon \frac{T_n}{b_{2T} d_{1T} m_{nT}} K_{Fa} K_{F\beta} K_{Fv}, \text{ МПа}, \quad (2.69)$$

где $Y_\beta = 1 - \frac{\beta}{140}$ – коэффициент наклона зубьев; Y_ε – коэффициент перекрытия зубьев, ориентировочно можно принять $Y_\varepsilon = 1$; $K_{F\alpha}$ – коэффициент распределения нагрузки между зубьями; $K_{F\beta}$ – коэффициент распределения нагрузки по ширине венца (см. табл. 2.13); K_{Fv} – коэффициент динамической нагрузки (см. табл. 2.14).

Значение $K_{F\alpha}$ определяют по формуле

$$K_{F\alpha} = \frac{4 + (\varepsilon_\alpha - 1)(n' - 5)}{4\varepsilon_\alpha}, \quad (2.70)$$

где n' – степень точности передачи.

Полученное значение σ_F не должно превышать $[\sigma_F]$ «слабого» звена более чем на 5%.

Таблица 2.13

Значения $K_{F\beta}$

Относительная ширина колеса Ψ_{bd}	Шестерня расположена симметрично относительно опор		Шестерня расположена несимметрично относительно опор				Консольное расположение шестерни или колеса	
			весьма жесткий вал		менее жесткий вал			
	твердость поверхностей зубьев HB_2		твердость поверхностей зубьев HB_2		твердость поверхностей зубьев HB_2		твердость поверхностей зубьев HB_2	
	> 350	< 350	> 350	< 350	> 350	< 350	> 350	< 350
0,2	1,00		1,02	1,01	1,10	1,05	1,25	1,13
0,4	1,03	1,01	1,07	1,04	1,20	1,12	1,55	1,28
0,6	1,05	1,02	1,13	1,07	1,30	1,17	1,90	1,50
0,8	1,08	1,05	1,20	1,11	1,44	1,23	2,30	1,70
1,0	1,10	1,04	1,18	1,08	1,37	1,15	–	
1,2	1,13	1,05	1,25	1,10	1,50	1,18		
1,4	1,15	1,07	1,32	1,13	1,60	1,23		
1,6	1,20	1,08	1,40	1,16	–	1,28		

2.3.5.10. Проверка тихоходной ступени на статическую прочность при перегрузках

Максимальное контактное напряжение под действием пикового крутящего момента определяют по формуле

$$\sigma_{H \max} = \sigma_H \sqrt{\frac{P_{дв}}{P_{тр}} \left(\frac{T_{\max}}{T} \right)}, \quad (2.71)$$

где $\left(\frac{T_{\max}}{T} \right)$ – заданная кратность пикового момента.

Таблица 2.14

Значения K_{Fv}

Степень точности	Твердость HB_2	Окружная скорость зуба $v_{п}$, м/с					
		1	2	4	6	8	10
6	≤ 350	1,02	1,04	1,07	1,10	1,15	1,18
	> 350	1,01	1,02	1,03	1,04	1,06	1,07
7	≤ 350	1,03	1,06	1,11	1,16	1,22	1,27
	> 350	1,01	1,02	1,03	1,05	1,07	1,08
8	≤ 350	1,03	1,06	1,11	1,17	1,23	1,29
	> 350	1,01	1,02	1,03	1,05	1,07	1,08
9	≤ 350	1,04	1,07	1,14	—		
	> 350	1,01	1,02	1,04			

Контактная прочность при перегрузках обеспечивается, если выполняется условие: $\sigma_{H \max} \leq [\sigma_{H \max}]$.

Максимальное напряжение изгиба под действием пикового крутящего момента определяют для «слабого» звена передачи по формуле

$$\sigma_{F \max} = \sigma_F \frac{P_{дв}}{P_{тр}} \left(\frac{T_{\max}}{T} \right). \quad (2.72)$$

Изгибная прочность при перегрузках обеспечивается, если выполняется условие: $\sigma_{F \max} \leq [\sigma_{F \max}]$.

2.3.5.11. Проектировочный расчет быстроходной ступени

Межосевое расстояние $a_{вб}$ определяют в зависимости от разновидности редуктора:

- в трехосном редукторе по $a_{вт}$ и табл. 2.9;
- в соосном редукторе $a_{вб} = a_{вт}$.

Геометрические параметры $m_{нб}$; $\beta_б$; $d_{1б}$; $d_{2б}$; $d_{a1б}$; $d_{a2б}$; $b_{2б}$; $b_{1б}$; числа зубьев $z_{1б}$ и $z_{2б}$, а также передаточное число ступени $u_{фб}$ определяют по формулам и рекомендациям, изложенным в подпункте 2.3.5.7.

В соосном редукторе значение $b_{2б}$ находят, пользуясь рекомендацией по $\psi_{баб}$ из табл. 2.3.

Фактическое передаточное число редуктора вычисляют по (2.73)

$$u_{ф} = u_{фб} u_{фт} \quad (2.73)$$

и проверяют выполнение условия (2.31).

2.3.5.12. Проверочные расчеты быстроходной ступени

Проверку быстроходной ступени на выносливость по контактными и изгибными напряжениями, а также на статическую прочность при перегрузках производят по формулам и рекомендациям, изложенным в подпунктах 2.3.5.8 – 2.3.5.10.

2.4. Конические зубчатые передачи

2.4.1. Геометрические параметры конических зубчатых передач

Наибольшее распространение получили **ортогональные** конические зубчатые передачи с углом между осями шестерни и колеса 90° .

По направлению зуба различают передачи **прямозубые** (зуб расположен по образующей конуса), с **тангенциальным зубом** (зуб расположен под углом к образующей конуса) и с **круговым зубом**.

В основном применяются передачи прямозубые и с круговым зубом, т.к. нагрузочная способность передач с тангенциальным зубом практически не выше, чем прямозубых.

Передачи с круговым зубом по сравнению с прямозубыми имеют, при равных нагрузках, на 15...20% меньшие габариты, работают более плавно и способны передавать вращение с большими окружными скоростями.

Недостатками передач с круговым зубом являются:

- большие величины осевых сил;
- зависимость направления осевой силы от направления вращения звена.

Эти свойства делают нежелательным применение передач с круговым зубом в преобразователях движения реверсивных ММ. Как правило, их используют **при постоянном направлении вращения** выходного звена, причем направление зуба назначают так, чтобы **осевые силы** действовали **к основаниям** образующих **конусов**.

На рис. 2.8 изображен фрагмент конической зубчатой передачи и показаны ее основные геометрические параметры.

Угол Σ между осями шестерни и колеса является одним из таких параметров. Ниже будут рассматриваться передачи, в которых $\Sigma = 90^\circ$.

Внешний окружной модуль прямых зубьев обозначается m_e , круговых – m_{te} . Далее обозначения параметров передач с круговым зубом даются в скобках.

Диаметры шестерни (индекс 1) и колеса (индекс 2):

внешние делительные

$$d_{e1} = m_e(m_{te})z_1; \quad (2.74)$$

$$d_{e2} = m_e(m_{te})z_2; \quad (2.75)$$

внешние окружностей вершин зубьев

$$d_{ae1} = d_{e1} + 2h_{ae1}\cos\delta_1; \quad (2.76)$$

$$d_{ae2} = d_{e2} + 2h_{ae2}\cos\delta_2; \quad (2.77)$$

внешние окружностей впадин

$$d_{fe1} = d_{e1} - 2h_{fe1}\cos\delta_1; \quad (2.78)$$

$$d_{fe2} = d_{e2} - 2h_{fe2}\cos\delta_2. \quad (2.79)$$

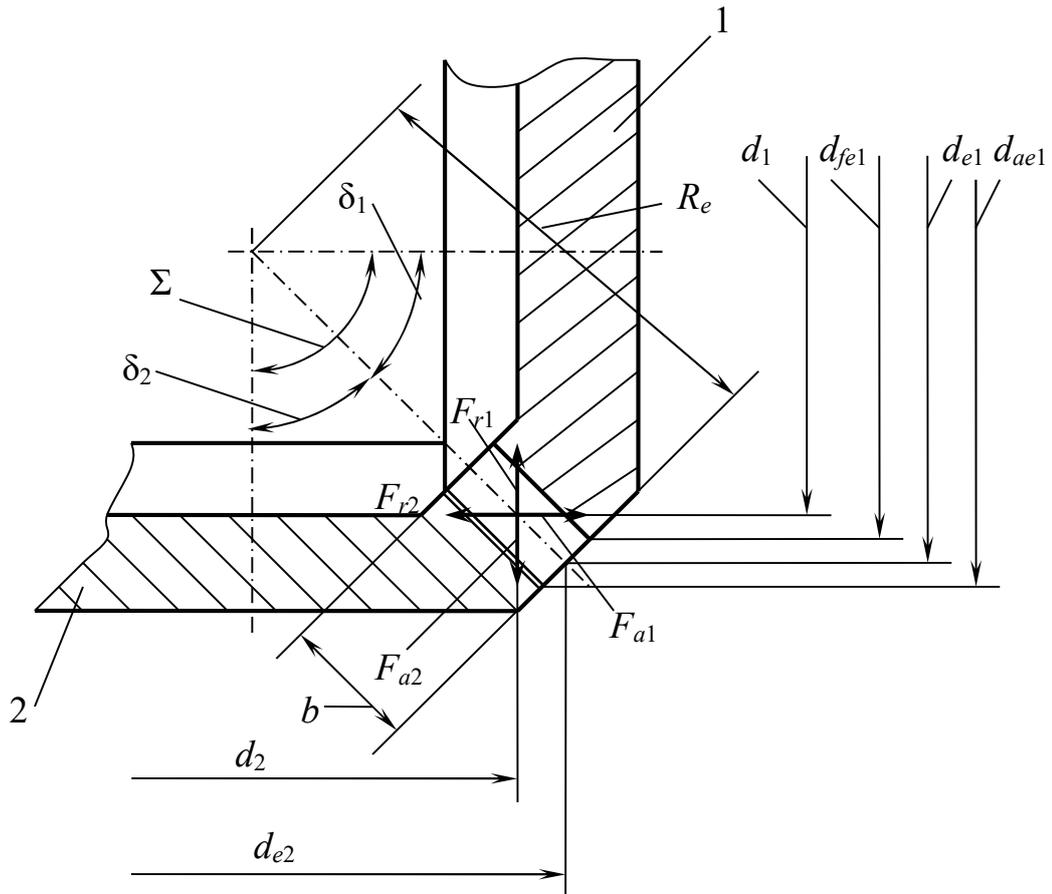


Рис. 2.8. Основные геометрические параметры конической зубчатой передачи

В формулах (2.74) – (2.79): z_1, z_2 – числа зубьев; h_{ae1}, h_{ae2} – высота головки зуба; h_{fe1}, h_{fe2} – высота ножки зуба; δ_1, δ_2 – углы делительных конусов.

Величины h_{ae}, h_{fe} находят с учетом коэффициентов смещения исходного контура $x_e(x_n)$ (подробнее см. [1]).

Значения δ_1, δ_2 :

$$\delta_1 = \operatorname{arctg} \frac{z_1}{z_2} = \operatorname{arctg} \frac{1}{u}; \quad (2.80)$$

$$\delta_2 = 90^\circ - \delta_1. \quad (2.81)$$

Внешнее конусное расстояние

$$R_e = 0,5m_e(m_{te})\sqrt{z_1^2 + z_2^2} = \frac{d_{e2}}{2\sin\delta_2}. \quad (2.82)$$

Ширину венца колеса b вычисляют по рекомендации $b = 0,285R_e$ и принимают ближайший размер по ряду R_{a40} .

В передаче с круговым зубом к основным параметрам относится также **угол наклона зуба к образующей конуса в среднем сечении $\beta = 35^\circ$** (рис. 2.9).

Средние делительные диаметры:

$$d_1 = d_{e1}\left(1 - \frac{b}{2R_e}\right) \approx 0,857d_{e1}; \quad (2.83)$$

$$d_2 = d_{e2}\left(1 - \frac{b}{2R_e}\right) \approx 0,857d_{e2}. \quad (2.84)$$

Кроме указанных выше параметров, используют также **средний окружной модуль**, определяемый в прямозубых передачах по выражению

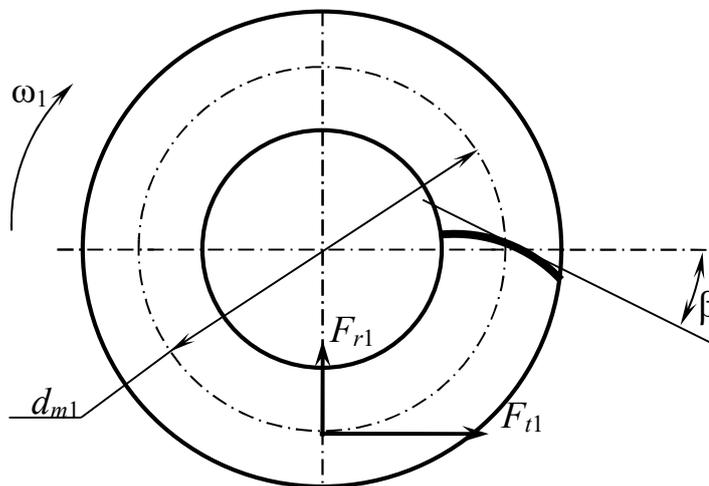


Рис. 2.9. Коническая шестерня с круговым зубом (вид со стороны вершины конуса)

$$m_m = \frac{d_1}{z_1} = \frac{d_2}{z_2} \approx 0,857m_e, \quad (2.85)$$

а в передачах с круговым зубом – по выражению

$$m_{nm} \approx 0,857m_{te} \cos \beta \approx 0,717m_{te}, \quad (2.86)$$

и среднее конусное расстояние

$$R_m = R_e - 0,5b = 0,9375R_e. \quad (2.87)$$

В передачах с круговым зубом необходимо правильно ориентировать зубья на звеньях в зависимости от направления вращения. Надлежащее расположение зуба шестерни при ее вращении по часовой стрелке показано на рис. 2.9.

2.4.2. Силы в конической зубчатой передаче

На рис. 2.8, 2.9 показаны направления сил в конической передаче. Окружная сила (см. рис. 2.9)

$$F_t = \frac{2T_2}{d_2}. \quad (2.88)$$

В прямозубой передаче (см. рис. 2.8):
радиальная сила на колесе, осевая на шестерне

$$F_{r2} = F_{a1} = F_t \operatorname{tg} \alpha_w \sin \delta_1 = 0,364F_t \sin \delta_1; \quad (2.89)$$

осевая сила на колесе, радиальная на шестерне

$$F_{a2} = F_{r1} = F_t \operatorname{tg} \alpha_w \cos \delta_1 = 0,364F_t \cos \delta_1. \quad (2.90)$$

В передаче с круговым зубом:
радиальная сила на колесе, осевая на шестерне

$$F_{r2} = F_{a1} = F_t (0,444 \sin \delta_1 + 0,701 \cos \delta_1); \quad (2.91)$$

осевая сила на колесе, радиальная на шестерне

$$F_{a2} = F_{r1} = F_t (0,444 \cos \delta_1 - 0,701 \sin \delta_1). \quad (2.92)$$

2.4.3. Методика расчета конической зубчатой передачи

2.4.3.1. Общие положения

Преобразователи движения ММ, предусматриваемых заданиями на курсовое проектирование, могут содержать коническую зубчатую передачу в виде либо одноступенчатого редуктора, либо ступени двухступенчатого редуктора. Далее рассматривается коническая передача как быстроходная ступень коническо-цилиндрического редуктора.

Общий порядок расчета – см. подпункт 2.3.5.1.

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3.

2.4.3.2. Кинематический расчет редуктора

Расчетное передаточное число редуктора определяют по (2.30) с проверкой по условию (2.31).

Распределение найденного u по ступеням необходимо выполнять с учетом следующего.

Характерной особенностью конических зубчатых передач является технологическая сложность нарезания зубьев колес при передаточных числах больших пяти. Поэтому в коническо-цилиндрических редукторах принимают $u_{\text{б}} \leq 5$, руководствуясь рекомендациями табл. 2.15.

Таблица 2.15

Передаточные числа ступеней коническо-цилиндрических редукторов

Передаточное число ступени	Передаточное число редуктора u										
	9	10	11,2	12,5	14	16	18	20	22,4	25	≤ 28
$u_{\text{б}}$	3,15	3,55		4	4,5				5		
$u_{\text{т}}$	2,8		3,15		3,55			4	4,5	5	$\leq 5,6$

Для конических зубчатых передач допускается отклонение $\Delta u = 4\%$, независимо от передаточного числа $u_{\text{б}}$.

Частоты вращения и угловые скорости валов определяют по (2.32) – (2.36).

2.4.3.3. Материалы шестерен и колес. Допускаемые напряжения

Материалы звеньев конической зубчатой передачи назначают аналогично цилиндрической передаче (см подпункт 2.3.5.5).

Допускаемые напряжения передачи прямозубой конической определяют так же, как прямозубой цилиндрической; передачи с круговым зубом – как косозубой цилиндрической.

Крутящие моменты на шестернях и колесах ступеней – см. подпункт 2.3.5.6.

2.4.3.4. Расчет тихоходной ступени

Проектировочный расчет тихоходной (цилиндрической) ступени – см. подпункт 2.3.5.7.

Проверочные расчеты – см. подпункты 2.3.5.8, 2.3.5.9.

2.4.3.5. Проектировочный расчет быстроходной (конической) ступени

Внешний делительный диаметр колеса равен

$$d_{e2} = 1650 \sqrt[3]{\frac{u_6 T_{II} K_{H\beta}}{\theta_H [\sigma_H]^2}}, \text{ мм}, \quad (2.93)$$

где θ_H – коэффициент вида конических колес.

Коэффициенты $K_{H\beta}$ для передач, валы которых установлены на роликовых подшипниках [5], с достаточной точностью могут быть определены по аппроксимирующим формулам:

прямозубые передачи при $HB_2 \leq 350$

$$K_{H\beta} = 0,22u_6; \quad (2.94)$$

прямозубые передачи при $HB_2 > 350$

$$K_{H\beta} = 0,29u_6; \quad (2.95)$$

передачи с круговым зубом при $HB_2 > 350$

$$K_{H\beta} = 0,24u_6. \quad (2.96)$$

В передачах с круговым зубом при $HB_2 \leq 350$ $K_{H\beta} = 1$.

Значение $\theta_H = 1$ для прямозубых колес; θ_H для колес с круговым зубом – по формулам [3]:

при $HB_1 \leq 350, HB_2 \leq 350$

$$\theta_H = 1,22 + 0,21u_6; \quad (2.97)$$

при $HB_1 > 350, HB_2 \leq 350$

$$\theta_H = 1,13 + 0,13u_6; \quad (2.98)$$

при $HB_1 > 350, HB_2 > 350$

$$\theta_H = 0,81 + 0,15u_6. \quad (2.99)$$

Значение d_{e2} следует принять из ряда R_{a40} ближайшее к полученному по (2.93).

Затем по (2.80) и (2.81) предварительно определяют углы δ_1 и δ_2 , после чего по (2.82) находят R_e .

Точность вычисления углов – до $0,0001^\circ$ (или до угловых секунд); R_e – до $0,001$ мм.

Ширину зубчатого венца находят по рекомендации, данной в пункте 2.4.1. Определяют внешний окружной модуль

$$m_e(m_{te}) = \frac{1,4 \cdot 10^4 T_n K_{F\beta}}{\theta_F d_{e2} b [\sigma_F]}, \text{ мм}, \quad (2.100)$$

где $K_{F\beta} = 1$ – для прямозубых передач; $K_{F\beta} = 1,08$ – для передач с круговым зубом; θ_F – коэффициент вида конических колес.

Значение $\theta_F = 0,85$ для прямозубых колес; θ_F для колес с круговым зубом – по формулам [6]:

при $HB_1 \leq 350, HB_2 \leq 350$

$$\theta_F = 0,94 + 0,08u_6; \quad (2.101)$$

при $HB_1 > 350, HB_2 \leq 350$

$$\theta_F = 0,85 + 0,04u_6; \quad (2.102)$$

при $HB_1 > 350, HB_2 > 350$

$$\theta_F = 0,65 + 0,11u_6. \quad (2.103)$$

Значение модуля до целого числа не округлять; точность – до $0,01$ мм. В силовых передачах рекомендуется принимать $m_e(m_{te}) \geq 1,5$ мм.

Числа зубьев:

$$z_{26} = \frac{d_{e2}}{m_e(m_{te})}; \quad (2.104)$$

$$z_{16} = \frac{z_{26}}{u_6}. \quad (2.105)$$

Полученные z_{16}, z_{26} округлить до целых.

Фактическое передаточное число пары $u_{ф6}$ найти по (2.10) и проверить на допустимость его отклонения от стандартного.

По (2.80) и (2.81) с использованием $u_{ф6}$ уточнить углы δ_1 и δ_2 .

После этого следует выбрать коэффициент смещения при нарезании зубьев шестерни $x_{e1}(x_{n1})$. При $HB_1 - HB_2 \leq 100$ его принимают по табл. 2.16. В случае отличия z_1 и $u_{\phi 6}$ от значений из табл. 2.16 $x_{e1}(x_{n1})$ принимают с округлением до табличных **в большую сторону**. При $HB_1 > 350, HB_2 > 350$ $x_{e1}(x_{n1}) = 0$, при $HB_1 - HB_2 > 100$ принимают $x_{n1} = 0$.

Коэффициент смещения при нарезании зубьев колеса $x_{e2}(x_{n2}) = -x_{e1}(x_{n1})$.

В заключение проектировочного расчета по (2.76), (2.77) найти d_{ae1} и d_{ae2} , а по (2.83), (2.84) – $d_{1\phi}$ и $d_{2\phi}$.

2.4.3.6. Проверка быстроходной ступени на выносливость по контактными напряжениями

Действительное контактное напряжение равно

$$\sigma_H = 470 \sqrt{\frac{F_t \sqrt{u_{\phi 6}^2 + 1}}{\theta_H d_{e2} b}} K_{H\beta} K_{Hv}, \text{ МПа.} \quad (2.106)$$

Таблица 2.16

Значения коэффициентов x_{e1}, x_{n1}

z_1	x_{e1} при передаточном числе u					x_{n1} при передаточном числе u				
	2,00	2,50	3,15	4,00	5,00	2,00	2,50	3,15	4,00	5,00
12	–	0,50	0,53	0,56	0,57	0,32	0,37	0,39	0,41	0,42
13	0,44	0,48	0,52	0,54	0,55	0,30	0,35	0,37	0,39	0,40
14	0,42	0,47	0,50	0,52	0,53	0,29	0,33	0,35	0,37	0,38
15	0,40	0,45	0,48	0,50	0,51	0,27	0,31	0,33	0,35	0,36
16	0,38	0,43	0,46	0,48	0,49	0,26	0,30	0,32	0,34	0,35
18	0,36	0,40	0,43	0,45	0,46	0,24	0,27	0,30	0,32	
20	0,34	0,37	0,40	0,42	0,43	0,22	0,26	0,28	0,29	
25	0,29	0,33	0,36	0,38	0,39	0,19	0,21	0,24	0,25	
30	0,25	0,28	0,31	0,33	0,34	0,16	0,18	0,21	0,22	
40	0,20	0,22	0,24	0,26	0,27	0,11	0,14	0,16	0,17	

Коэффициент K_{Hv} для прямозубых передач определяется по табл. 2.17, а для передач с круговым зубом – по табл. 2.11 в зависимости от степени точности, которую находят по рекомендациям для цилиндрических передач и окружной скорости

$$v_n = \frac{\omega_n d_{2\phi}}{2000}, \text{ м/с.} \quad (2.107)$$

Таблица 2.17

Значения K_{Hv} для прямозубых конических передач

Степень точности	Твердость HB_2	Окружная скорость зуба $v_{п}$, м/с					
		1	2	4	6	8	10
6	≤ 350	1,03	1,06	1,12	1,17	1,23	1,28
	> 350	1,02	1,04	1,07	1,10	1,15	1,18
7	≤ 350	1,04	1,07	1,14	1,21	1,29	1,36
	> 350	1,03	1,05	1,09	1,14	1,19	1,24
8	≤ 350	1,04	1,08	1,16	1,24	–	
	> 350	1,03	1,06	1,10	1,16		
9	≤ 350	1,05	1,10	–			
	> 350	1,04	1,07				

Полученное значение контактного напряжения проверяют на соответствие условию (2.68). В случае невыполнения условия следует изменить размер b . Если увеличение (или уменьшение) b на два соседних размера по ряду R_a40 не дает достаточного эффекта, необходимо перейти на другой размер d_{e2} или назначить другие материалы передачи.

2.4.3.7. Проверка быстроходной ступени на выносливость по напряжениям изгиба

Напряжения изгиба в зубьях колеса и шестерни определяют по следующим формулам:

$$\sigma_{F2} = Y_{F2} Y_{\beta} \frac{F_{t6} K_{F\beta} K_{Fv}}{\theta_F b m_e (m_{te})}; \quad (2.108)$$

$$\sigma_{F1} = \sigma_{F2} \frac{Y_{F1}}{Y_{F2}}, \quad (2.109)$$

где Y_{F1} , Y_{F2} – по табл. 2.18; Y_{β} – для прямозубых передач равен единице, для передач с круговым зубом равен 0,75; K_{Fv} – для прямозубых передач – по табл. 2.19, для передач с круговым зубом – по табл. 2.14.

Полученные значения σ_{F1} , σ_{F2} не должны превышать соответствующие допускаемые напряжения более чем на 5%. При невыполнении этого условия следует увеличить модуль и, оставив без изменения d_{e2} , пересчитать числа зубьев колеса и шестерни.

2.4.3.8. Проверочные расчеты быстроходной ступени на статическую прочность при перегрузках

Проверочные расчеты производят по формулам и рекомендациям, изложенным в подпункте 2.3.5.10.

Таблица 2.18

Значения коэффициента Y_F для конических колес

z_v	Значения $x_e(x_n)$										
	-0,5	-0,4	-0,3	-0,2	-0,1	0	0,1	0,2	0,3	0,4	0,5
12	–								3,90	3,67	3,46
14	–						4,24	4,00	3,78	3,59	3,42
17	–				4,50	4,27	4,03	3,83	3,67	3,53	3,40
20	–			4,55	4,28	4,07	3,89	3,75	3,61	3,50	3,39
25	–	4,60	4,39	4,20	4,04	3,90	3,77	3,67	3,57	3,46	3,39
30	4,60	4,32	4,15	4,05	3,90	3,80	3,70	3,62	3,55	3,47	3,40
40	4,12	4,02	3,92	3,84	3,77	3,70	3,64	3,58	3,53	3,48	3,42
50	3,97	3,88	3,81	3,76	3,70	3,65	3,61	3,57	3,53	3,49	3,44
60	3,85	3,79	3,73	3,70	3,66	3,63	3,59	3,56	3,53	3,50	3,46
80	3,73	3,70	3,68	3,65	3,62	3,61	3,58	3,56	3,54	3,52	3,50
100	3,68	3,67	3,65	3,62	3,61	3,60	3,58	3,57	3,55	3,53	3,52
Эквивалентные числа зубьев: $z_{v1} = z_1 \cos \delta_1$; $z_{v2} = z_2 \cos \delta_2$											

Таблица 2.19

Значения K_{Fv} для прямозубых конических передач

Степень точности	Твердость HB_2	Окружная скорость зуба v_n , м/с					
		1	2	4	6	8	10
6	≤ 350	1,05	1,06	1,12	1,17	1,20	1,25
	> 350	1,02	1,04	1,08	1,11	1,14	1,17
7	≤ 350	1,08	1,16	1,33	1,50	1,67	1,80
	> 350	1,03	1,05	1,09	1,13	1,17	1,22
8	≤ 350	1,10	1,20	1,38	1,58	1,78	1,96
	> 350	1,04	1,06	1,12	1,16	1,21	1,26
9	≤ 350	1,13	1,28	1,50	–		
	> 350	1,04	1,07	1,14	–		

2.5. Червячные передачи

2.5.1. Геометрические параметры червячных передач

Червячные передачи предназначены для преобразования вращательного движения между двумя **скрещивающимися осями** (межосевой угол в плане обычно равен 90°).

Различают передачи с **цилиндрическим червяком** (архимедовым, конволютным, эвольвентным) и с **глобоидным червяком**. Более высокой нагрузочной способностью обладают передачи с глобоидным червяком [7], однако они значительно сложнее в изготовлении, поэтому используются только в обоснованных случаях. В основном же в ММ и роботах применяют передачи с

цилиндрическим червяком (рис. 2.10), из которых, в свою очередь, наибольшее распространение получили преобразователи движения с **архимедовым червяком**. Архимедов червяк в сечении, проходящем через продольную ось, выглядит как зубчатая рейка (рис. рис. 2.11).

Фактическое передаточное число червячной передачи (т.е. передаточное отношение от червяка к червячному колесу) равно

$$u_{\phi} = \frac{z_2}{z_1}, \quad (2.110)$$

где z_1 – **число витков червяка**.

Стандартные значения z_1 : 1; 2; 4. В обоснованных случаях в преобразователях движения ММ и роботов могут быть заданы и другие числа витков.

Обычно u лежит в пределах 8...80, но в отдельных случаях может значительно превышать верхнее значение.

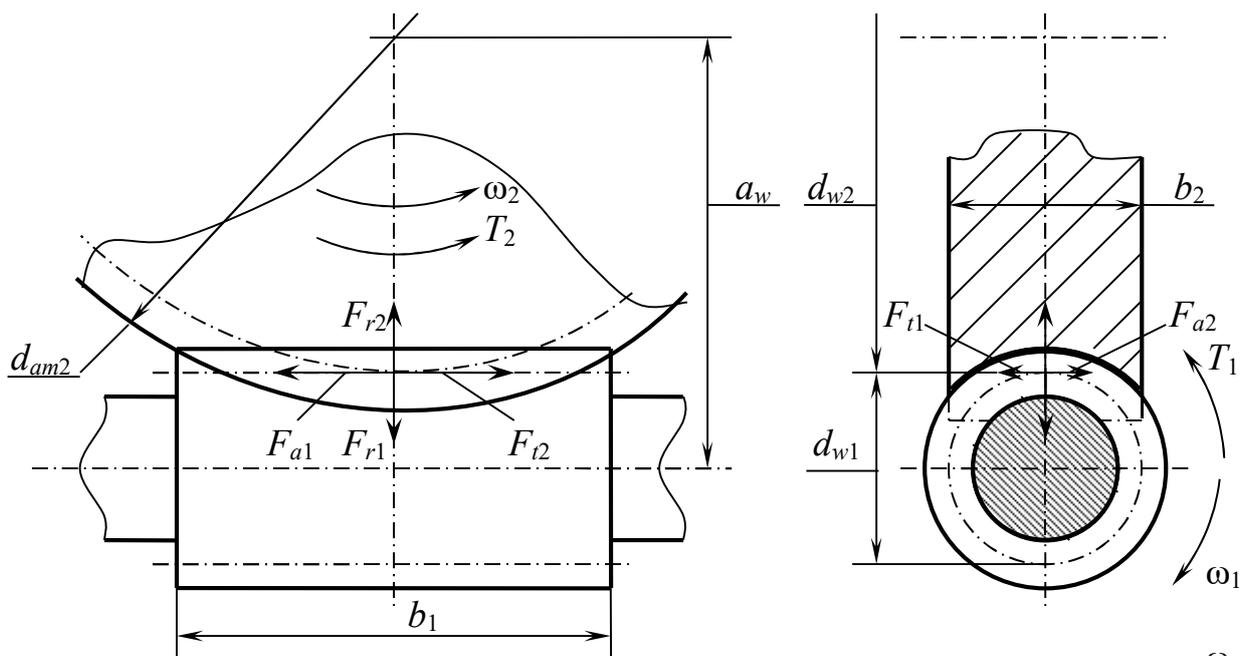


Рис. 2.10. Схема червячной передачи с цилиндрическим червяком

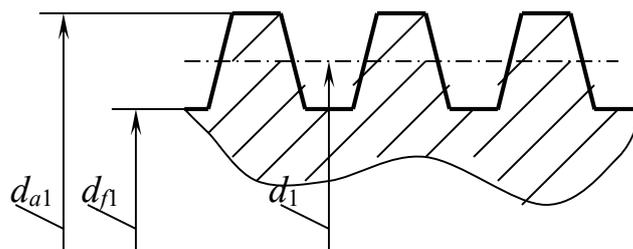


Рис. 2.11. Осевое сечение витков архимедова червяка

Диаметр делительной окружности червяка (рис. 2.11)

$$d_1 = qm, \quad (2.111)$$

где q – коэффициент диаметра, принимается из стандартного ряда.

Модуль m также стандартизован. Ряды значений q и m даны в п. 2.5.4.

Межосевое расстояние a_w (рис. 2.10) равно

$$a_w = 0,5m(q + z_2 + 2x), \quad (2.112)$$

где x – коэффициент смещения, его величина находится в пределах: $-1 \leq x \leq 1$.

Диаметр вершин витков и диаметр впадин червяка

$$d_{a1} = (q + 2)m; \quad (2.113)$$

$$d_{f1} = (q - 2,4)m. \quad (2.114)$$

Диаметр делительной окружности колеса

$$d_2 = mz_2;$$

диаметр вершин зубьев и диаметр впадин колеса

$$d_{a2} = d_2 + 2(1 + x)m; \quad (2.115)$$

$$d_{f2} = d_2 - 2(1,2 - x)m; \quad (2.116)$$

наибольший диаметр колеса

$$d_{am2} = d_{a2} + \frac{6m}{z_1 + 2}. \quad (2.117)$$

Диаметр начальной окружности червяка

$$d_{w1} = (q + 2x)m; \quad (2.118)$$

диаметр начальной окружности колеса $d_{w2} = d_2$.

Длина нарезаемой части червяка

$$b_1 = (10 + 5,5|x| + z_1)m + C, \quad (2.119)$$

где $C = 0$ при $x \leq 0$; при $x > 0$

$$C = \frac{100m}{z_2}. \quad (2.120)$$

Ширина венца колеса:
при $z_1 = 1; z_1 = 2$

$$b_2 = 0,355a_w; \quad (2.121)$$

при $z_1 = 4$

$$b_2 = 0,315a_w. \quad (2.122)$$

2.5.2. Силы в червячной передаче

Силу нормального давления на зуб колеса **в передаче с архимедовым червяком** (по аналогии с косозубой цилиндрической передачей) можно представить в виде геометрической суммы трех составляющих (рис. 2.10):
окружной силы

$$F_{t2} = \frac{2T_2}{d_2};$$

радиальной силы

$$F_{r2} = F_{t2} \operatorname{tg} \alpha_w = 0,364 F_{t2}; \quad (2.123)$$

осевой силы

$$F_{a2} = F_{t1}, \quad (2.124)$$

где T_2 – крутящий момент на колесе; $\alpha_w = 20^\circ$ – угол зацепления; F_{t1} – окружная сила на червяке, равная

$$F_{t1} = \frac{2T_1}{d_1}; \quad (2.125)$$

T_1 – крутящий момент на червяке.

Кроме окружной, на червяк действуют радиальная $F_{r1} = F_{r2}$ и осевая $F_{a1} = F_{t2}$ силы.

КПД червячной пары определяют по формуле

$$\eta_{\text{ч}} = \frac{\operatorname{tg} \gamma}{\operatorname{tg}(\gamma + \varphi')}, \quad (2.126)$$

где $\gamma = \operatorname{arctg} \frac{z_1}{q}$ – угол подъема витка червяка; φ' – угол трения (см. п. 2.5.4).

2.5.3. Материалы червячных передач

Червяки передач малой и средней мощности, работающих с большими перерывами и редко испытывающих перегрузки, выполняют из сталей марок 40Х, 35ХМ, 40ХН. Витки таких червяков упрочняют закалкой с нагревом ТВЧ до 45...50HRC. Поверхности витков шлифуют.

Червяки тяжело нагруженных ответственных передач выполняют цементованными с закалкой до 56...63HRC с последующей шлифовкой и полировкой витков. Часто используются недорогие цементуемые стали, например, сталь 18ХГТ.

Материалы венцов червячных колес приведены в табл. 2.20. Группа материала назначается по табл. 2.21 в зависимости от **скорости скольжения**

$$v_s = 4 \cdot 10^{-4} n_1 \sqrt[3]{T_2}, \text{ м/с}, \quad (2.127)$$

где n_1 – частота вращения червяка, об/мин.

Материалы III группы в преобразователях движения ММ и роботов применяются редко, поэтому далее рассматриваться не будут.

2.5.4. Методика расчета червячной передачи

2.5.4.1. Общие положения

Преобразователи движения ММ, предусматриваемых заданиями на курсовое проектирование, могут содержать червячную передачу в виде либо одноступенчатого редуктора, либо ступени двухступенчатого редуктора. Далее рассматривается одноступенчатый червячный редуктор.

Общий порядок расчета – см. подпункт 2.3.5.1.

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3, причем требуемую номинальную мощность двигателя вычисляют по формуле

$$P_{\text{тр}} = \frac{T_2 n_2}{\eta_{\text{ч}} \eta_{\text{п}}}, \quad (2.128)$$

где $\eta_{\text{ч}}$ предварительно принимают равным 0,8.

2.5.4.2. Кинематический расчет передачи

Расчетное передаточное число редуктора определяют по (2.30) с проверкой по условию (2.31). В качестве $n_{\text{т}}$ используют n_2 . Допускаемое отклонение $\Delta u = 4\%$.

Таблица 2.20

Материалы венцов червячных колес

Группа	Марка материала	Способ отливки	Механические свойства, МПа	
			σ_B	σ_T
Ia	БрО10Н1Ф1	Ц	285	165
	БрО10Ф1	К	275	200
		З	230	140
Iб	БрО5Ц5С5	К	200	90
		З	145	80
IIa	БрА10Ж4Н4	Ц	700	460
		К	650	430
	БрА10ЗМц1,5	К	550	360
		З	450	300
	БрФ9ЖЗЛ	Ц	530	245
		К	500	230
З		425	195	
IIб	ЛЦ23А6ЖЗМц2	Ц	500	330
		К	450	295
		З	400	260
III	СЧ18	З	355	-
	СЧ15	З	315	-

Способы отливки: Ц – центробежный; К – в кокиль; З – в землю

Таблица 2.21

Выбор группы материалов венцов червячных колес

Характеристика передачи	Скорость скольжения v_s , м/с			
	до 2	2...3	3...4	более 4
Слабо нагруженная, режим работы легкий	III	IIб	-	
Средне и тяжело нагруженные, режимы работы средний и тяжелый	IIб	IIa	Iб	Ia
Тяжело нагруженная, режим работы весьма тяжелый	IIa	Iб	Ia	

Определяют z_1 по рекомендации: при $u \geq 31,5$ $z_1 = 1$; при $u = 16...28$ $z_1 = 2$; при $u = 8...14$ $z_1 = 4$.

Определяют z_2 по формуле

$$z_2 = \frac{n_{дв}}{z_1} = \frac{n_1}{z_1} \quad (2.129)$$

с округлением до целого числа, после чего по (2.110) находят u_{ϕ} и снова проверяют выполнение условия (2.31). В случае его невыполнения z_2 изменяют на один зуб в большую или меньшую сторону.

Вычисляют частоты вращения и угловые скорости червяка и червячного колеса.

2.5.4.3. Материалы червяка и колеса. Допускаемые напряжения

По (2.127) определяют ориентировочное значение скорости скольжения, по табл. 2.21 – группу материала, а по табл. 2.20 – марку материала венца колеса.

По рекомендациям п. 2.5.3 принимают марку стали и термообработку витков червяка.

Вычисляют коэффициенты долговечности:

$$K_{HL} = \sqrt[8]{\frac{10^7}{N_{HE}}}; \quad (2.130)$$

$$K_{FL} = \sqrt[9]{\frac{10^6}{N_{FE}}}. \quad (2.131)$$

В (2.130), (2.131) при $N_{HE} > 25 \cdot 10^7$ принимают $N_{HE} = 25 \cdot 10^7$, при $N_{FE} > 25 \cdot 10^7$ принимают $N_{FE} = 25 \cdot 10^7$, при $N_{FE} < 10^6$ принимают $N_{FE} = 10^6$.

По табл. 2.22 принимают коэффициент износа материала C_v , по табл. 2.23 устанавливают формулы и находят допускаемые напряжения для червячного колеса.

Таблица 2.22

Значения коэффициента износа материала

$v_s, \text{ м/с}$	1	2	3	4	5	6	7	≥ 8
C_v	1,33	1,21	1,11	1,02	0,95	0,88	0,83	0,8

Таблица 2.23

Допускаемые напряжения для червячного колеса

Группа материалов	Червяк < 45HRC	Червяк \geq 45HRC	Нереверсивная передача	Реверсивная передача
	[σ] _H , Н /мм ²			
I	$K_{HL}C_v0,75\sigma_B$	$K_{HL}C_v0,9\sigma_B$	$(0,08\sigma_B+0,25\sigma_T)K_{FL}$	$0,16\sigma_BK_{FL}$
II	$250 - 25v_s$	$300 - 25v_s$		

2.5.4.4. Проектировочный расчет передачи
Межосевое расстояние определяют по формуле

$$a_w = 610 \sqrt[3]{\frac{T_2}{[\sigma_H]^2}} \quad (2.132)$$

где T_2 – в ньютонметрах.

Полученное значение a_w округляют до ближайшего по ряду $R_a 20$.

Определяют модуль зацепления

$$m = (1,5 \dots 1,7) a_w / z_2, \text{ мм}, \quad (2.133)$$

и принимают ближайшее стандартное значение из 1-го ряда: 2,50; 3,15; 4,00; 5,00; 6,30; 8,00; 10,00; 12,50; 16,00. Допускается также использование 2-го ряда: 3,00; 3,50; 6,00; 7,00; 12,00.

Коэффициент диаметра определяют по рекомендации

$$q = (0,212 \dots 0,250) z_2. \quad (2.134)$$

Полученную величину округляют до стандартной из 1-го ряда: 6,3; 8,0; 10,0; 12,5; 16,0 или из 2-го ряда: 7,1; 9,0; 11,2; 14,0; 18,0. По ГОСТ 19672-74 допускается также применение значений q 7,5 и 12,0.

Коэффициент смещения нарезающего инструмента находят из выражения

$$x = (a_w / m) - 0,5(q + z_2) \quad (2.135)$$

с проверкой на соответствие критерию из п. 5.2.1.

По (2.112) определяют фактическое межосевое расстояние.

Вычисляют основные геометрические размеры звеньев передачи по формулам (2.113) – (2.122).

Кроме того, находят делительный угол подъема витков

$$\gamma = \arctg(z_1 / q). \quad (2.136)$$

2.5.4.5. Проверочные расчеты

Уточняют КПД передачи по формуле (2.126), причем угол трения φ' принимают по табл. 2.24 в зависимости от фактической скорости скольжения

$$v_{s\varphi} = \frac{u_\varphi \omega_2 d_1}{2000 \cos \gamma}, \text{ м/с}. \quad (2.137)$$

Таблица 2.24

Значения угла трения

$v_{s\phi}$	0,1	0,5	1	1,5	2	2,5	3	4	7	10	15
φ'	4°30'... ...5°10'	3°10'... ...3°40'	2°30'... ...3°10'	2°20'... ...2°50'	2°00'... ...2°30'	1°40'... ...2°20'	1°30'... ...2°00'	1°20'... ...1°40'	1°00'... ...1°30'	0°55'... ...1°20'	0°50'... ...1°10'
Меньшие значения φ' – для материалов группы I, большие – для материалов групп II и III.											

В том случае, если получается $\eta_{\phi} < 0,8$, следует заново подсчитать требуемую мощность электродвигателя и при необходимости назначить двигатель более мощный.

Уточняют значение $[\sigma]_H$ по фактической скорости скольжения и формулам из табл. 2.23.

Контактное напряжение в зацеплении сравнивают с допусковым:

$$\sigma_H = 340 \sqrt{\frac{F_{t2} K}{d_1 d_2}} \leq [\sigma]_H, \quad (2.138)$$

где K – коэффициент нагрузки (принимается в зависимости от окружной скорости колеса (формула (2.139)): при $v_2 \leq 3$ м/с $K = 1$; при $v_2 > 3$ м/с $K = 1,1 \dots 1,3$).

$$v_2 = \frac{\omega_2 d_2}{2000}, \text{ м/с.} \quad (2.139)$$

Допускаемая недогрузка передачи составляет 15%, допускаемая перегрузка – 5%.

Напряжение изгиба в зубе колеса сравнивают с допусковым:

$$\sigma_F = 0,7 \frac{Y_{F2} F_{t2} K}{b_2 m} \leq [\sigma]_F, \quad (2.140)$$

где Y_{F2} – коэффициент формы зуба колеса, принимается по табл. 2.25 в зависимости от эквивалентного числа зубьев $z_{v2} = z_2 / \cos^3 \gamma$.

Таблица 2.25

Коэффициент формы зуба червячного колеса

z_{v2}	Y_{F2}	z_{v2}	Y_{F2}	z_{v2}	Y_{F2}	z_{v2}	Y_{F2}
20	1,98	30	1,76	40	1,55	80	1,34
24	1,88	32	1,71	45	1,48	100	1,30
26	1,85	35	1,64	50	1,45	150	1,27
28	1,80	37	1,61	60	1,40	300	1,24

Как правило, получается $\sigma_F \ll [\sigma]_F$, так как нагрузочная способность червячных передач ограничивается не изгибной, а контактной выносливостью.

Выполняют проверку (тепловой расчет) редуктора на нагрев. Цель проверки – определить температуру масла в редукторе, которая не должна превышать допускаемую $[t]_M = 80 \dots 95^\circ$.

Температуру масла в корпусе червячной передачи при непрерывной работе находят по формуле

$$t_M = t_B + \frac{P_1(1 - \eta_\phi)}{K_t A}, \quad (2.141)$$

где t_B – температура окружающего воздуха, принимают $t_B = 20^\circ\text{C}$; P_1 – мощность на червяке, Вт; K_t – коэффициент теплопередачи (среднее значение коэффициента $K_t = 13 \text{ Вт}/(\text{м}^2 \cdot \text{град})$); A – площадь теплоотдающей поверхности корпуса (табл. 2.26), м^2 .

Таблица 2.26

Площадь теплоотдающей поверхности корпуса червячного редуктора

Межосевое расстояние, мм	80	100	125	140	160	180	200	224
Площадь A	0,19	0,24	0,36	0,43	0,56	0,67	0,8	1

При невыполнении условия $t_M \leq [t]_M$ следует увеличить с помощью ребрения площадь поверхности теплоотдачи. Если этой меры недостаточно, то необходимо предусмотреть специальные средства охлаждения (обдув корпуса вентилятором, введение в конструкцию холодильника для масла).

2.6. Планетарные зубчатые передачи

2.6.1. Общие сведения о планетарных передачах

Конструкции планетарных передач весьма многообразны. Простейшая планетарная передача (рис. 2.12) состоит из центрального **солнечного** зубчатого колеса 1 с наружными зубьями, центрального **корончатого** зубчатого колеса 3 с внутренними зубьями, **сателлитов** 2 с внешними зубьями, которые входят в зацепление одновременно с солнечным и корончатым колесами, и **води́ла** H , на котором расположены оси сателлитов.

При закреплённом корончатом колесе 3 ($\omega_3 = 0$) вращение солнечного колеса с угловой скоростью ω_1 вызывает вращение сателлитов относительно собственных осей с угловой скоростью ω_2 , что вызывает качение сателлитов по корончатому колесу и приводит к их перемещению по круговой орбите радиуса R_H , а следовательно, и водила H с угловой скоростью ω_H .

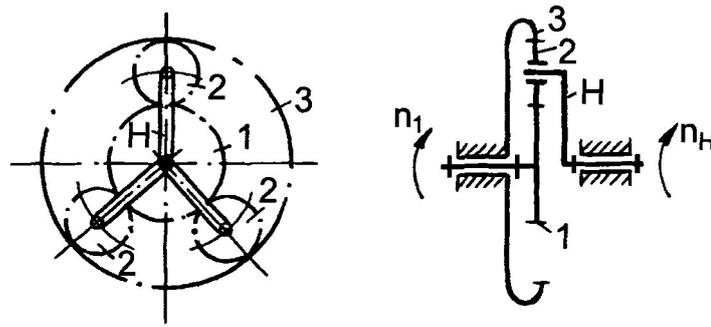


Рис. 2.12. Планетарная передача схемы 2К-Н

Сателлиты вращаются относительно водила со скоростью $\omega_2^H = \omega_2 - \omega_H$ и вместе с водилом совершают переносное движение. Их движения напоминают движения планет, поэтому передачу и называют планетарной.

Основными звеньями планетарной передачи являются те, которые воспринимают внешние моменты. Такими звеньями являются солнечные и корончатые колеса, т.е. два центральных колеса (2К) и водило Н. Сокращенное обозначение такого планетарного механизма «2К-Н».

Планетарный механизм 2К-Н наиболее часто используется в ММ, так как имеет высокий КПД и технологичную конструкцию. Поэтому ниже будет дана методика расчета именно этой схемы.

Подробно и полно планетарные передачи рассмотрены в специальной литературе [7]. Здесь же следует сказать о весьма перспективной конструкции – так называемом эксцентриково-планетарном механизме, вариант которого – **механизм с параллельными кривошипами** – приведен на рис. 2.13.

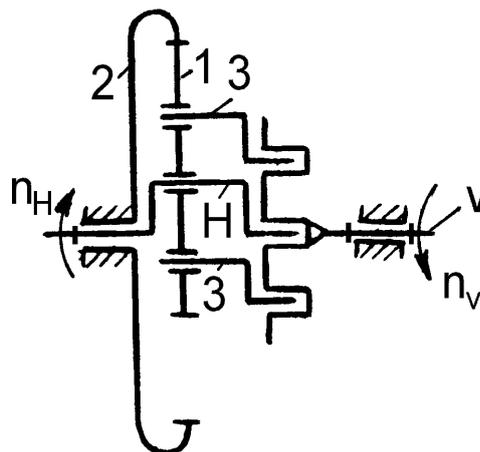


Рис. 2.13. Планетарный механизм с параллельными кривошипами (схема К-Н-В)

Планетарный механизм К-Н-В имеет три основных звена: центральное колесо 2, водило Н и вал V. Водило представляет собой эксцентрик, на котором

установлен сателлит 1. Сателлит передает вращение на вал V кривошипами 3 – эксцентриками, эксцентриситет которых равен эксцентриситету водила.

За один оборот водила сателлит, совершая сложное плоское движение, поворачивается относительно своей оси на угол, соответствующий разности чисел зубьев центрального колеса и сателлита, в направлении, противоположном направлению вращения водила, т. е. передаточное число такого преобразователя движения равно

$$u^{(2)}_{H-V} = -z_2 / (z_2 - z_1). \quad (2.142)$$

Для предотвращения заклинивания зубьев в зацеплении должно выполняться условие

$$z_2 - z_1 \geq \begin{cases} 8 \text{ при } h_a^* = 1; \\ 7 \text{ при } h_a^* = 0,8, \end{cases} \quad (2.143)$$

где h_a^* – высота головки зуба.

Очевидно, что при достаточно больших числах зубьев звеньев этой пары одна ступень передачи $K-H-V$ может по передаточному числу заменить двухступенчатый цилиндрический редуктор.

Дальнейшие исследования в направлении уменьшения размеров и повышения технологичности изготовления передачи привели к появлению так называемого **планетарно-цевочного редуктора** (рис. 2.14), в котором вместо эвольвентного зацепления применяют цевочное.

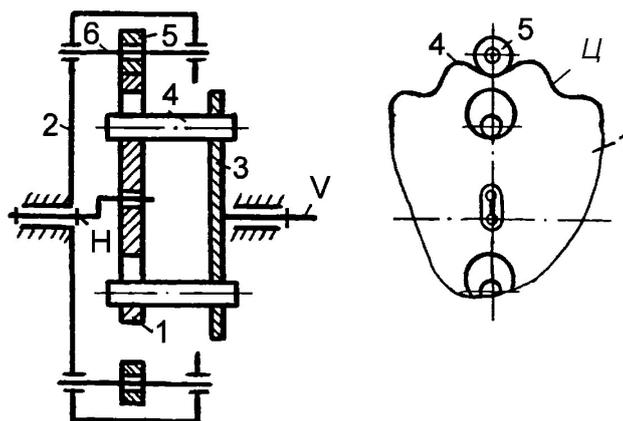


Рис. 2.14. Планетарно-цевочный редуктор схемы $K-H-V$

Сателлит 1 имеет зубья, выполненные по циклоиде (Ц), и смонтирован на водиле H . Центральное колесо 2 содержит оси 6 с установленными на них с возможностью вращения цевками 5, взаимодействующими с зубьями сателлита. Выходной вал V соединен с диском 3, в отверстия которого с натягом вставлены

пальцы 4, входящие в отверстия сателлита. Пальцы, контактируя со стенками отверстий, выполняют функцию параллельных кривошипов в схеме по рис. 2.13. Далее будет рассмотрена планетарная передача, показанная на рис. 2.12.

2.6.2. Передаточное число и условия существования планетарного механизма

Передаточное число передачи $2K-H$ равно

$$u_{1H}^{(3)} = 1 + \frac{z_3}{z_1}, \quad (2.144)$$

где $u_{1H}^{(3)}$ – передаточное число от солнечного колеса к водилу при неподвижном солнечном колесе.

При проектировании следует учитывать, что передаточное число данного механизма не должно превышать 8, в крайнем случае – 9. При необходимости обеспечения большего передаточного числа преобразователь движения выполняют многоступенчатым, причем каждая ступень представляет собой передачу по рис. 2.12.

Числа зубьев колес выбирают так, чтобы отсутствовали подрезание и заклинивание зубьев. Для этого число зубьев солнечного колеса должно удовлетворять следующим рекомендациям: $z_1 \geq 24$ при выполнении солнечного колеса из стали нормализованной и улучшенной твердостью $HB \leq 350$; $z_1 \geq 21$ – из стали закаленной ТВЧ твердостью $HRC \leq 52$; $z_1 \geq 18$ – из стали цементованной твердостью $HRC \geq 52$. Минимальное число зубьев корончатого колеса

$$z_{2\min} \geq \begin{cases} 85 & \text{при } h_a^* = 1; \\ 58 & \text{при } h_a^* = 0,8. \end{cases} \quad (2.145)$$

Для существования планетарного механизма необходимо, чтобы числа зубьев соответствовали условиям сборки, соосности и соседства сателлитов.

Из условия соосности следует, что число зубьев сателлита должно быть равно

$$z_2 = \frac{z_3 - z_1}{2}. \quad (2.146)$$

По условию сборки необходимо, чтобы выполнялось равенство

$$\frac{z_3 + z_1}{C} = \gamma, \quad (2.147)$$

где C – число сателлитов (обычно $C = 3$); γ – целое число.

Условие соседства сателлитов выглядит следующим образом:

$$z_2 + 2 \leq (z_1 + z_3) \cdot \sin \frac{\pi}{C}. \quad (2.148)$$

Следует также иметь в виду, что числа зубьев колес z_1 и z_3 должны быть или оба четные, или оба нечетные.

2.6.3. Материалы планетарных передач

Для изготовления звеньев планетарных передач используют те же марки сталей и методы термообработки, что и для цилиндрических зубчатых передач, с учетом следующих рекомендаций.

Так как зуб солнечного колеса более часто входит в зацепление, чем зуб сателлита, то при твердости зубьев сателлита $HВ \leq 350$ твердость зубьев солнечного колеса назначают на 50...70 единиц выше, чем сателлита. При твердости зубьев сателлита $HВ > 350$ твердости солнечного колеса и сателлита назначают одинаковыми.

Для изготовления водил используют углеродистые стали. Корпуса планетарных передач ММ обычно изготавливают из легких материалов и сплавов.

2.6.4. Методика расчета планетарной передачи

2.6.4.1. Общие положения

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3.

2.6.4.2. Кинематический расчет передачи

Цель кинематического расчета – определение чисел зубьев солнечного и корончатого колес и сателлита, обеспечивающих требуемое передаточное число, а также число сателлитов.

Находят расчетное передаточное число

$$u_{1H\Phi}^{(3)} = \frac{n_1}{n_H}, \quad (2.149)$$

где n_H – заданная частота вращения водила.

Определение чисел зубьев колес планетарных передач производят обычно методом подбора, задаваясь числом зубьев солнечного колеса по рекомендациям п. 2.6.2 и обеспечивая при этом правильность зацепления.

Задавшись z_1 , предварительно находят z_3 по формуле

$$z_3 = z_1(u_{1Hp}^{(3)} - 1). \quad (2.150)$$

Затем из условия соосности вычисляют число зубьев сателлита z_2 и проверяют выполнение условий сборки и соседства сателлитов. При невыполнении какого-либо из условий изменяют числа зубьев колес на $\pm 1...3$ зуба и добиваются выполнения условия.

После определения чисел зубьев находят фактическое передаточное число

$$u_\phi = 1 + \frac{z_3}{z_1}. \quad (2.151)$$

Вычисляют отклонение фактического передаточного числа от расчетного. Допускаемое отклонение передаточного отношения обычно принимают равным 4 %.

При невыполнении этого условия необходимо число z_1 уменьшить, снова найти числа z_2 и z_3 , после чего провести проверку механизма по условиям соосности, сборки и соседства сателлитов.

2.6.4.3. Материалы и допускаемые напряжения

Материалы звеньев передачи назначают с учетом рекомендаций, приведенных в п. 2.6.3.

Допускаемые напряжения в парах центральное колесо – сателлит и сателлит – корончатое колесо определяют так же, как в цилиндрических зубчатых передачах.

2.6.4.4. Проектировочный расчет передачи

Межосевое расстояние пары солнечное колесо – сателлит

$$a_w = K_a (u + 1) \sqrt[3]{\frac{T_1 K_{H\beta} K_c}{[\sigma]_H^2 u^2 C \psi_{ba}}}, \text{ мм}, \quad (2.152)$$

где u – передаточное число пары солнечное колесо – сателлит (отношение чисел зубьев большего колеса к числу зубьев меньшего); T_1 – вращающий момент на солнечном колесе, Нм; K_c – коэффициент неравномерности распределения нагрузки между сателлитами (при наличии механизма выравнивания нагрузки $K_c = 1,1...1,2$; при отсутствии – $K_c = 1,5...2,0$); C – число сателлитов; ψ_{ba} – коэффициент ширины зубчатого венца сателлита.

Коэффициент ψ_{ba} равен

$$\psi_{ba} = b_C/a_w, \quad (2.153)$$

где b_C – ширина венца сателлита.

Значение ψ_{ba} принимают из ряда стандартных значений: 0,100; 0,105; 0,160; 0,200; 0,250; 0,315; 0,400; 0,500; 0,630; 0,800 (желательно принимать $\psi_{ba} = 0,400$ для материалов колес твердостью $HВ \leq 350$; $\psi_{ba} = 0,315$ при твердости $HRC \leq 50$; $\psi_{ba} = 0,250$ при твердости $HRC > 50$).

Затем находят делительный диаметр меньшего из колес пары солнечное колесо – сателлит. Пусть меньшее – сателлит, тогда

$$d_C = \frac{2a_w}{u+1}. \quad (2.154)$$

Определяют модуль зубьев

$$m = \frac{d_1}{z_1} = \frac{d_C}{z_C}. \quad (2.155)$$

Полученное значение модуля округляют до стандартного, уточняют значения делительных диаметров солнечного колеса, сателлита, корончатого колеса, а также межосевого расстояния.

Затем по приведенным выше формулам вычисляют диаметры окружностей вершин и впадин зубьев всех колес передачи.

Определяют окружную скорость солнечного колеса и по ее величине назначают степень точности планетарной передачи по аналогии с цилиндрической зубчатой передачей.

Для мехатронных модулей рекомендуют 6 или 7 степени точности планетарных передач.

2.6.4.5. Проверочные расчеты

Проверку зубьев в паре солнечное колесо – сателлит на выносливость по контактным напряжениям производят в соответствии с формулой

$$\sigma_H = 22,4Z_H Z_M Z_\varepsilon \sqrt{\frac{T_1(u+1)^3 K_{H\alpha} K_{H\beta} K_{H\nu}}{b_C u C}} \leq [\sigma]_H, \quad (2.156)$$

где коэффициенты Z_M , Z_H , Z_ε определяют так же, как при расчете

цилиндрической зубчатой передачи.

При невыполнении условия (2.157) в расчет следует ввести поправку путем изменения ширины венца сателлита.

Проверку зубьев пары солнечное колесо – сателлит на выносливость по напряжениям изгиба производят для того колеса, чье отношение $\frac{[\sigma]_F}{Y_F}$ меньше, по аналогии с цилиндрическими зубчатыми передачами.

Допустимые отклонения фактических напряжений от допускаемых те же, что и для цилиндрических зубчатых передач.

Проверочный расчет зубьев колес планетарной передачи при перегрузках аналогичен такому же расчету цилиндрической зубчатой передачи.

2.7. Волновые зубчатые передачи

2.7.1. Общие сведения

Конструкция волновой зубчатой передачи поясняется рис. 2.15. По своей сути она представляет собой планетарную передачу, у которой одно из колес выполнено в виде **гибкого венца**.

Волновая зубчатая передача состоит из **гибкого колеса 1** с наружными зубьями, **жесткого колеса 2** с внутренними зубьями и **генераторам волн 3**.

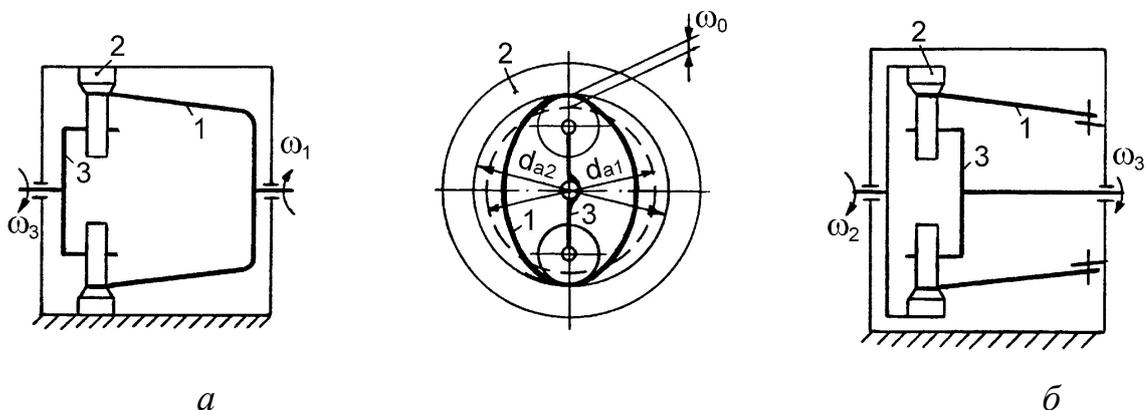


Рис. 2.15. Схема волновой зубчатой передачи

Наружный диаметр d_{a1} гибкого колеса в недеформированном состоянии меньше внутреннего диаметра d_{a2} жесткого колеса на величину

$$d_{a2} - d_{a1} = 2\omega_0, \quad (2.157)$$

где ω_0 – размер, показанный на рис. 2.15.

Возможны исполнения передачи с ведомым гибким колесом (рис. 2.15, а) и с ведомым жестким колесом (рис. 2.15, б).

О конструкции гибкого колеса будет сказано ниже.

Используются генераторы волн двух типов – кулачковый и дисковый.

Кулачковый генератор волн (рис. 2.16) содержит кулачок 1, рабочая поверхность которого спрофилирована по эллипсу, и размещенный на нем гибкий шариковый подшипник 2.

Параметры гибких подшипников указаны в табл. 2.27.

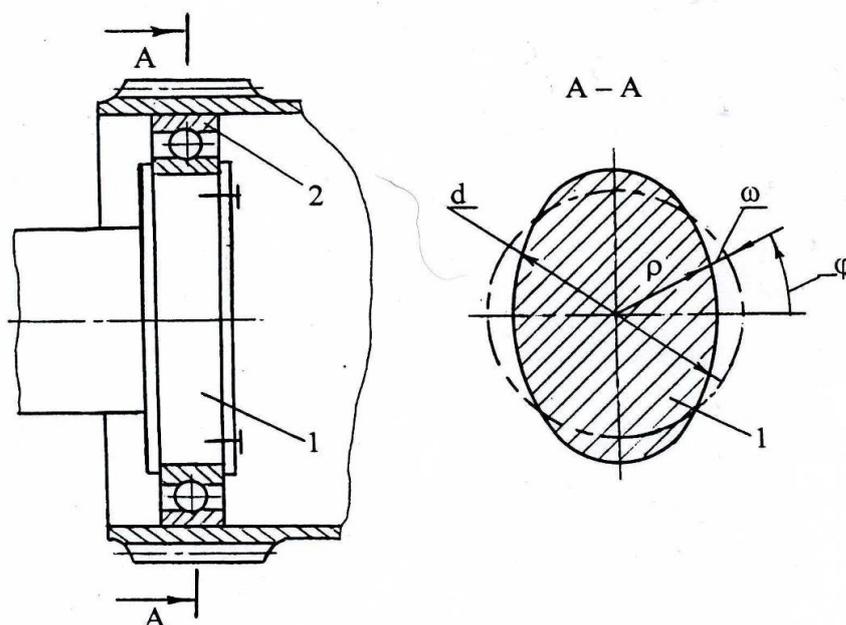


Рис. 2.16. Кулачковый генератор волн

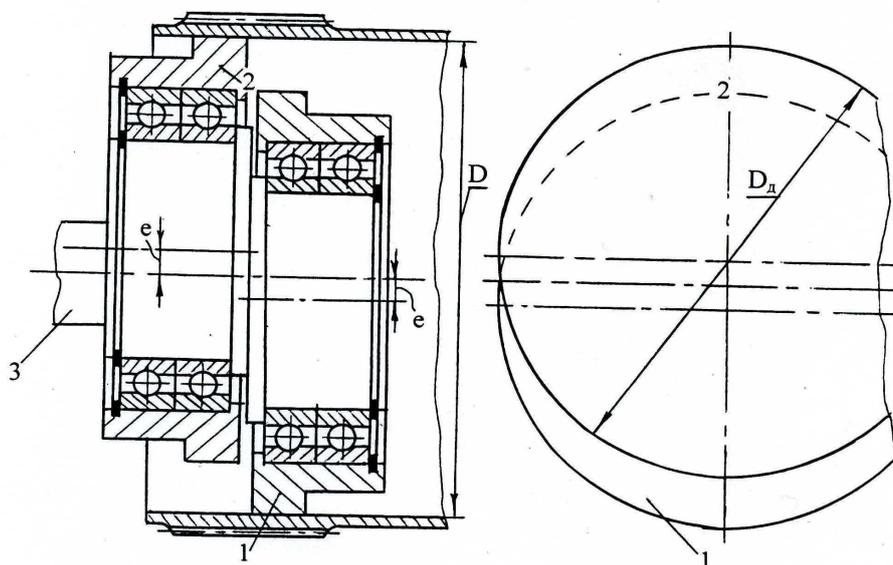


Рис. 2.17. Дисковый генератор волн

Дисковый генератор волн (рис. 2.17) содержит диски 1 и 2, расположенные относительно быстроходного вала 3 с эксцентриситетом e посредством шариковых подшипников.

Диапазон передаточных чисел волновых зубчатых передач: $80 \leq u \leq 400$.

Вследствие многопарности зацепления для этих передач характерна весьма высокая кинематическая точность. При сравнительно малых габаритах и массе волновые передачи способны создавать на выходном звене вращающие моменты до 150 кНм. На волновом принципе основаны механизмы для передачи вращения в герметично закрытое пространство.

Недостатки волновых передач:

- сложность изготовления;
- пониженный по сравнению с цилиндрическими зубчатыми передачами КПД ($\eta_{\text{вп}} = 0,8 \dots 0,9$);
- ограниченность частоты вращения кулачкового генератора волн максимальной частотой вращения гибкого подшипника;
- высокая чувствительность к качеству смазки.

2.7.2. Передаточное число волновой передачи

При неподвижном жестком зубчатом колесе 2 (рис. 2.15, а) передаточное число от вала генератора волн 3 к валу гибкого колеса 1 равно

$$u_{31}^{(2)} = -\frac{n_6}{n_T} = -\frac{\omega_3}{\omega_1} = -\frac{z_1}{z_2 - z_1} = -\frac{z_1}{K_z v} = -\frac{d_1}{d_2 - d_1}. \quad (2.158)$$

Знак минус указывает на разные направления вращения ведущего и ведомого звеньев.

Передаточное число от вала генератора волн 3 к валу жесткого колеса 2 при неподвижном гибком колесе 1 (рис. 2.15, б) равно

$$u_{32}^{(1)} = \frac{n_6}{n_T} = \frac{\omega_3}{\omega_2} = \frac{z_2}{z_2 - z_1} = \frac{z_2}{K_z v} = \frac{d_2}{d_2 - d_1}. \quad (2.159)$$

В формулах (2.159), (2.160): $\omega_1, \omega_2, \omega_3$ – угловая скорость соответственно гибкого колеса, жесткого колеса и генератора волн; z_1, z_2 – числа зубьев гибкого и жесткого колес; d_1, d_2 – диаметры делительных окружностей гибкого и жесткого колес; v – число волн деформации ($v = 1, 2, 3 \dots$, обычно $v = 2$, реже 3); K_z – коэффициент кратности (обычно $K_z = 1$; при $u < 70$ $K_z = 2$; при $u < 45$ $K_z = 3$).

2.7.3. Геометрические параметры волновой передачи

Основные геометрические параметры волновой передачи показаны на рис. 2.18 (буквами а, б, в на рис. 2.18 обозначены варианты конструктивного исполнения гибкого колеса).

Подшипники шариковые однорядные для волновых передач (ГОСТ23179-78)

Обозначение подшипника	Размеры, мм					Число шариков $z_{Ш}$	n_{max} , об/мин
	d	D	B	r	$d_{Ш}$		
806	30	42	7	1,0	3,969	21	3000
808	40	52	8		23		
809	45	62	9		5,953	21	
812	60	80	13		7,144	23	
815	75	100	15		9,128	21	
818	90	120	18		11,113	23	
822	110	150	24		14,288	21	1500
824	120	160					
830	150	200	30		19,050		
836	180	240	35		1,5	22,225	
844	220	300	45	2,5	28,575		
848	240	320	48	2,5	36,513	23	1000
860	300	400	60				
862	310	420	70				
872	360	480	72	3,5	44,450		

d – внутренний диаметр подшипника; D – наружный диаметр подшипника; B – ширина подшипника; r – радиус скругления кромок колец; $d_{Ш}$ – диаметр шариков.

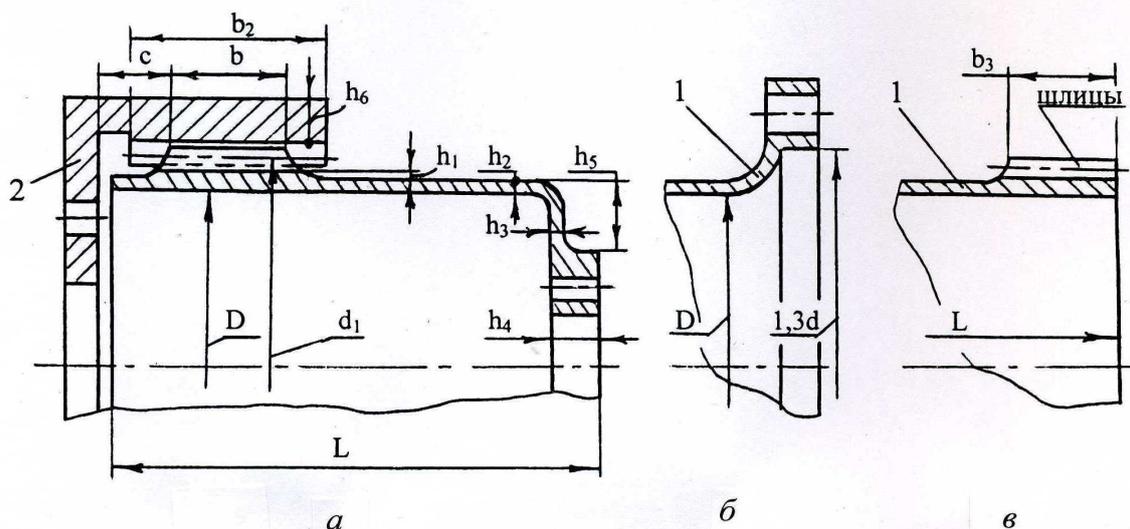


Рис. 2.18. Геометрические параметры волновой передачи

Параметры гибкого колеса:

- внутренний диаметр D ;
- делительный диаметр зубчатого венца d_1 ;

- ширина зубчатого венца гибкого колеса b_1 ;
- толщина зубчатого венца гибкого колеса h_1 ;
- толщина оболочки гибкого колеса $h_2 = h_3$;
- длина гибкого колеса L ;
- конструктивные размеры – b_3 ; h_4 ; h_5 .

Параметры жесткого колеса:

- делительный диаметр зубчатого венца d_2 ;
- ширина зубчатого венца гибкого колеса b_2 ;
- конструктивные размеры – c ; h_6 .

Важным геометрическим параметром зацепления является модуль m , величина которого оговаривается стандартным рядом, мм: 0,20; 0,22; 0,28; 0,30; 0,35; 0,40; 0,45. Ряд содержит также значения меньше 0,20 мм, однако их применяют редко из-за сложности нарезания внутренних зубьев жесткого колеса.

2.7.4. Материалы волновых передач

В качестве материалов зубчатых колес, как правило, используют стали, которые после термообработки имеют твердость 220...330 *НВ* (табл. 2.28). В отдельных случаях колеса изготавливают из бериллиевой бронзы БрБ2 или пластмассы.

Таблица 2.28

Марки сталей для изготовления зубчатых колес волновых передач

Сталь	Твердость, <i>НВ</i>	Предел выносливости σ_{-1} , МПа
40Х	280...300	500
40ХНМА	310...330	550
30ХГСА	300...320	530
38ХНЗВА	310...330	550
38ХМЮА	220...240	550
ШХ15	260...280	420
Х18Н10Т	220...240	350

2.7.5. Методика расчета волновой передачи

2.7.5.1. Общие положения

Исходные данные для расчета – см. подпункт 2.3.5.2.

Выбор двигателя – см. подпункт 2.3.5.3.

2.7.5.2. Кинематический расчет передачи

Кинематический расчет волновой передачи заключается в подборе чисел зубьев гибкого и жесткого колес по принятому в соответствии с исходными данными передаточным числом (формулы (2.159), (2.160)).

Число зубьев z_1 гибкого колеса определяют в зависимости от того, какое колесо вращается. При подвижном гибком колесе

$$z_1 = uK_z v, \quad (2.160)$$

при подвижном жестком колесе

$$z_1 = (u - 1)K_z v. \quad (2.161)$$

2.7.5.3. Материалы и допускаемое напряжение

Материал зубчатых колес принимают в соответствии с п. 2.7.4.

Допускаемые напряжения смятия на боковых поверхностях зубьев определяют в виде:

$$[\sigma]_{\text{см}} = 15,7K_u K_n K_{d1}, \text{ МПа}, \quad (2.162)$$

где $K_u = \frac{u - 20}{u}$ – коэффициент передаточного числа; $K_n = \frac{10}{\sqrt[3]{n_6}}$ – коэффициент

частоты вращения генератора волн (в понижающей передаче частота вращения генератора волн равна частоте быстроходного вала n_6); K_{d1} – коэффициент делительного диаметра гибкого колеса (при $d_1 \leq 130$ мм $K_{d1} = 1,25$; при $d_1 > 130$ мм $K_{d1} = 1$).

Для стальных зубчатых колес диапазон допускаемых напряжений составляет $[\sigma]_{\text{см}} = 10 \dots 20$ МПа, для пластмассовых – $[\sigma]_{\text{см}} = 3 \dots 5$ МПа.

2.7.5.4. Проектировочный расчет волновой передачи

Для выполнения расчета применяют упрощенные экспериментально проверенные зависимости. Они верны только для эвольвентных зубчатых колес, нарезанных стандартным инструментом с исходным контуром, имеющим угол зацепления $\alpha = 20^\circ$, коэффициент высоты зуба $h_a^* = 1$, коэффициент радиального зазора $C^* = 0,25$ для модулей более 1 мм и $C^* = 0,35$ для модулей до 1 мм.

Делительный диаметр гибкого колеса, мм

$$d_1 = \sqrt[3]{\frac{10T_1 K}{\Psi_{bd} [\sigma]_{\text{см}}}}, \quad (2.163)$$

где T_T – в ньютонметрах; K – коэффициент режима работы передачи (при спокойной нагрузке с кратностью перегрузки $T_{\max}/T \leq 1,2$ $K = 1$; при умеренной динамической нагрузке с $T_{\max}/T \leq 1,6$ $K = 1,25$; при резко динамической нагрузке с $1,6 < T_{\max}/T \leq 2,5$ $K = 1,75$); ψ_{bd} – коэффициент ширины зубчатого венца (для силовых ММ $\psi_{bd} = 0,15 \dots 0,20$; для малонагруженных кинематических ММ $\psi_{bd} = 0,06 \dots 0,15$).

Ширина зубчатого венца гибкого колеса равна

$$b_1 = \psi_{bd} d_1. \quad (2.164)$$

где ψ_{bd} – коэффициент ширины зубчатого венца (для силовых передач $\psi_{bd} = 0,15 \dots 0,20$; для малонагруженных кинематических передач $\psi_{bd} = 0,06 \dots 0,15$).

При кулачковом генераторе волн делительный диаметр предварительно принимают равным внутреннему диаметру D гибкого колеса, который считают приблизительно равным наружному диаметру гибкого подшипника (см. табл. 2.27).

Число зубьев z_1 :

при подвижном гибком колесе

$$z_1 = u K_z v, \quad (2.165)$$

при подвижном жестком колесе

$$z_1 = (u - 1) K_z v. \quad (2.166)$$

Модуль зубьев равен

$$m = d_1 / z_1. \quad (2.167)$$

После вычисления его по (2.168) необходимо принять ближайшее стандартное значение.

Уточняют делительный диаметр гибкого колеса в передаче с **дисковым генератором волн**:

$$d_1 = m z_1. \quad (2.168)$$

Находят фактическое передаточное число передачи в зависимости от того, какое колесо вращается:

при подвижном гибком колесе

$$u_{\phi} = z_1 / (K_z v);$$

при подвижном жестком колесе

$$u_{\phi} = z_1 / (K_z v) + 1. \quad (2.169)$$

Вычисляют отклонение фактического передаточного числа от требуемого и проверяют соответствие отклонения техническим требованиям к ММ. В случае сверхнормативного отклонения необходимо изменить z_1 и скорректировать значение d_1 .

Число зубьев жесткого колеса

$$z_2 = z_1 + K_z v. \quad (2.170)$$

Толщина зубчатого венца гибкого колеса

$$h_1 = (70 + 0,5u)mz_1 \cdot 10^{-4}, \text{ мм}. \quad (2.171)$$

Толщина оболочки гибкого колеса

$$h_2 = (0,5 \dots 0,8)h_1. \quad (2.172)$$

Относительный боковой зазор между зубьями ненагруженной передачи

$$\bar{j}_{max} = \frac{j_{max}}{m} = \frac{T_{max} b}{d_1^2 h_2 G m} + 4 \cdot 10^{-4} (u - 60), \quad (2.173)$$

где j_{max} – необходимый боковой зазор между зубьями ненагруженной передачи, мм; G – модуль упругости второго рода для материала гибкого колеса (для стали $G = 8,1 \cdot 10^4$ МПа).

Относительное радиальное упругое деформирование гибкого колеса

$$w_0^* = w_0 / m = 0,89 + 8 \cdot 10^{-4} z_1 + 2j_{max}, \quad (2.174)$$

где w_0 – радиальное упругое деформирование гибкого колеса, мм.

Коэффициент смещения исходного контура зубьев гибкого колеса

$$x_1 = \frac{1,35 - w_0^*}{\sqrt[3]{z_1} - 0,04}. \quad (2.175)$$

Коэффициент смещения исходного контура зубьев жесткого колеса

$$x_2 = x_1 + w_0^* - 1. \quad (2.176)$$

Относительная глубина захода зубьев

$$h_d^* = h_d/m = 4w_0^* - (4,6 - 4w_0^*) \cdot 10^{-3}z_1 - 2,48, \quad (2.177)$$

где h_d – глубина захода зубьев, мм.

Диаметр окружности впадин зубьев гибкого колеса

$$d_{f1} = m(z_1 - 2h_d^* - 2c^* + 2x_1). \quad (2.178)$$

Диаметр окружности вершин зубьев гибкого колеса

$$d_{a1} = d_{f1} + 2(h_d^* + c^*)m. \quad (2.179)$$

Диаметр окружности вершин зубьев жесткого колеса

$$d_{a2} = d_{a1} - 2(h_d^* - w_0^*)m. \quad (2.180)$$

Диаметр окружности впадин зубьев жесткого колеса

$$d_{f2} = d_{a1} + 2(0,15 + w_0^*)m. \quad (2.181)$$

Для гибкого колеса передачи с кулачковым генератором волн уточняют значение толщины венца по формуле

$$h_1 = 0,5(d_{f1} - D). \quad (2.182)$$

Длина гибкого колеса с дном (см. рис. 2.18, а) $L = 0,8D$, со шлицами (рис. 2.18, в) $L = 0,7D$.

Параметры остальных частей гибкого колеса: $h_3 = (0,7 \dots 1)h_2$; $h_4 = 2h_1$; $h_5 \geq 0,16D$; $c = 0,2b$.

Ширина зубчатого венца жесткого колеса

$$b_2 = b + 0,6\sqrt{b}. \quad (2.183)$$

Длина шлицев (рис. 2.18, в) $b_3 = 0,5b$.

Определяют основные геометрические параметры генератора волн.

Геометрию кулачкового генератора (см. рис. 2.16) определяет радиус-вектор поперечного сечения ρ .

Радиус-вектор кулачка в каждой четверти равен

$$\rho = 0,5d_{\pi} + WmK_{\alpha}, \quad (2.185)$$

где $d_{\text{п}}$ – внутренний диаметр гибкого подшипника; W – коэффициент радиальной деформации (табл. 2.29); K_{α} – коэффициент, зависящий от угла зацепления α ($K_{\alpha} = 1$ для $\alpha = 20^{\circ}$; $K_{\alpha} = 0,89$ для $\alpha = 30^{\circ}$).

Срок службы стандартных гибких подшипников составляет 10 000 ч, при эксплуатации допускаются кратковременные двукратные перегрузки.

Таблица 2.29

Значения коэффициента W

Угол ϕ	0	10°	20°	30°	40°	50°	60°	70°	80°	90°
W	-1,25	-1,13	-0,91	-0,56	-0,15	0,26	0,57	0,76	0,87	0,9

Эксцентриситет дискового генератора волн (см. рис. 2.17) равен

$$e = 3,4 w_0^* m. \quad (2.185)$$

Диаметр дисков

$$D_{\text{д}} = D + 2w_1 - 2e, \quad (2.186)$$

где w_1 – максимальная упругая деформация гибкого колеса с учетом податливости генератора волн и жесткого колеса, а также отклонений размеров от номинальных при изготовлении.

Значение w_1 определяют по формуле

$$w_1 = w_0^* m (0,97 + 0,025 \sqrt{\sigma_{\text{см}}}), \text{ мм.} \quad (2.187)$$

2.7.5.5. Проверочные расчеты

Запас усталостной прочности гибкого колеса

$$s = \frac{0,286 \sigma_{-1} u_{\phi} d_1}{K_d K_{\sigma} w_0^* E h_1 \sqrt{1 + 0,15 \left(\frac{d_1}{L} + \frac{T_{\tau} u_{\phi}}{w_0^* d_1 E h_1^2} \right)}}, \quad (2.188)$$

где $E = 2,1 \cdot 10^5$ МПа – модуль упругости первого рода для стали; σ_{-1} – предел выносливости материала гибкого колеса (см. табл. 2.28); K_d – коэффициент увеличения напряжения от сил в зацеплении (формула (2.191)); K_{σ} – эффективный коэффициент концентрации напряжений у основания зуба (формула (2.192)).

Условие обеспечения необходимой выносливости гибкого колеса

$$s \geq [s], \quad (2.189)$$

где $[s] = 1,2$ – минимальный допускаемый запас усталостной прочности.

Величину коэффициента K_d определяют по формуле

$$K_d = 1 + \frac{2,2T_r}{10^8 d_1 h_1^2}, \quad (2.190)$$

а коэффициента K_σ – по формуле

$$K_\sigma = 1 + \frac{0,05 \sqrt{\frac{m d_1 (h_a^* + c^*)}{R_{\min} h_1}} (R_{\min} + 0,01)}{R_{\min} + 0,02}, \quad (2.191)$$

где R_{\min} – минимальный радиус переходной поверхности, равный

$$R_{\min} = \frac{m(h_a^* + c^* - x_1 - \rho^*)}{h_a^* + c^* - x_1 - \rho^* + 0,5z_1} + m\rho^*. \quad (2.192)$$

Значения входящих в (2.193) величин: $c^* = 0,25$ и $\rho^* = 0,4$ при $m > 1$ мм; $c^* = 0,35$ и $\rho^* = 0,4$ при $m = 1 \dots 0,5$ мм; $c^* = 0,5$ и $\rho^* = 0,33$ при $m < 0,5$ мм.

Вероятность P неразрушения гибкого зубчатого колеса в зависимости от коэффициента запаса усталостной прочности приведена в табл. 3.20.

Таблица 2.30

Вероятность P неразрушения гибкого зубчатого колеса и коэффициент K_L вероятности неразрушения гибкого подшипника

s	1,80	1,70	1,60	1,55	1,50	1,45	1,40	1,30	1,20
$P, \%$	99,8	99,6	99,0	98,5	97,8	96,7	95,1	90,0	87,0
K_L	0,50	0,60	0,66	0,69	0,73	0,80	0,90	1,00	

Уточняют КПД передачи. Потери в волновой передаче складываются из потерь в зацеплении и потерь при вращении генератора волн. Сопротивлением подшипников быстроходного и тихоходного валов пренебрегают. В этом случае КПД равен

$$\eta = [1 + 1,11vf_1/\cos^2\alpha + 0,3uf_2\text{tg}(\alpha + \rho')]^{-1}, \quad (2.193)$$

где $f_1 = 0,03 \dots 0,05$ – коэффициент, учитывающий трение в зубчатом зацеплении; $f_2 = 0,0015 \dots 0,003$ – условный коэффициент, учитывающий трение во всех элементах генератора волн; $\rho' = \arctg f_1$ – приведенный угол трения.

После уточнения КПД выполняют **тепловой расчет** передачи. Температуру масла в корпусе редуктора определяют по формуле (2.141), причем коэффициент продолжительности работы редуктора находят по формуле

$$K_t = t_p / t_{\text{ц}}, \quad (2.194)$$

где t_p – время работы редуктора в течение цикла; $t_{\text{ц}}$ – длительность цикла, а поверхность охлаждения редуктора принимают равной $A = (8...10)d_1^2$, м². Полученную температуру сравнивают с допусковой.

3. ПОДШИПНИКИ

3.1. Классификация подшипников по виду трения

Подшипники служат опорами валов, вращающихся осей и других деталей, совершающих вращательное движение. Они воспринимают нагрузки, приложенные к вращающимся деталям, и сохраняют заданное положение осей вращения. От качества подшипников во многом зависит работоспособность и долговечность ММ и роботов. При выборе подшипников следует принимать во внимание, что они относятся к наименее долговечным узлам механизмов. Часто именно необходимость технического обслуживания или замены подшипников является причиной остановки ММ.

По виду трения различают **подшипники скольжения** и **подшипники качения**. Кроме того, подшипники скольжения разделяются на опоры **жидкостного, полужидкостного и полусухого трения**. При жидкостном трении рабочие поверхности деталей разделены слоем смазки, толщина которого больше суммы высот шероховатости поверхностей. Это условие не соблюдается в опорах полужидкостного и полусухого трения, поэтому вращение деталей в них сопровождается износом поверхностей даже без попадания абразивных частиц извне.

Практически во всех ответственных узлах ММ и роботов используются подшипники качения, что обусловлено следующими их **преимуществами** перед подшипниками скольжения:

- условный коэффициент трения подшипников качения мал, он равен 0,0015...0,0060 и приближается к коэффициенту жидкостного трения (0,001...0,005);

- подшипники качения не требуют большого количества смазки, что позволяет значительно упростить смазочную систему;

- конструкции подшипников качения позволяют производить их в массовых масштабах и обеспечивать их взаимозаменяемость.

К **недостаткам** подшипников качения следует отнести:

- отсутствие конструкций, разъемных в радиальном направлении;

- сравнительно большие радиальные габариты;

- ограниченную быстроходность, связанную с кинематикой и динамикой тел качения;

- низкую работоспособность в агрессивных средах, а также при вибрационных и ударных нагрузках.

Несмотря на сокращение области применения подшипников скольжения, в ряде случаев они остаются незаменимыми, и конструктор обязан это учитывать в процессе разработки ММ.

Далее будут рассмотрены подшипники качения.

3.2. Конструкции и классификация подшипников качения

В общем случае подшипник качения состоит из **колец** с беговыми дорожками, **тел качения** и **сепаратора**. Тела качения расположены между кольцами и перемещаются по беговым дорожкам. Сепаратор представляет собой элемент, охватывающий тела качения и распределяющий их равномерно по окружности.

Кроме того, конструкция подшипника может включать в себя другие детали: уплотнительные и защитные шайбы, конические втулки с круглыми гайками и др. В некоторых преобразователях движения используют, наоборот, упрощенные подшипники (без одного из колец). Со всем многообразием подшипников качения можно ознакомиться с помощью специальной технической литературы [5, 8].

По виду тел качения различают **шариковые** и **роликовые** подшипники. Роликовые делятся на подшипники с **короткими цилиндрическими роликами**, с **длинными цилиндрическими роликами**, **игольчатые**, с **коническими роликами** и с **бочкообразными роликами**.

По направлению воспринимаемой нагрузки подшипники подразделяются на **радиальные**, **упорные**, **радиально-упорные** и **упорно-радиальные**.

Радиальные шариковые подшипники – наиболее простые и дешевые. Они допускают небольшие перекосы вала (до $0,25^\circ$) и могут воспринимать не только радиальные, но и значительные осевые нагрузки. Благодаря своим достоинствам эти подшипники широко распространены в преобразователях движения ММ.

Радиальные роликовые подшипники с короткими цилиндрическими роликами способны воспринимать значительно бóльшие радиальные нагрузки, чем шариковые, но не допускают радиальных нагрузок и перекосов вала.

Радиально-упорные шариковые подшипники имеют особую форму наружного кольца, вследствие чего равнодействующая сил давления шарика на кольцо образует с диаметральной плоскостью угол α , называемый **углом контакта**.

В радиально-упорных и упорно-радиальных роликовых подшипниках используются конические ролики. Сравнение этих подшипников с радиально-упорными шариковыми показывает, что роликовые обладают существенно бóльшей нагрузочной способностью, но гораздо хуже воспринимают перекосы вала.

Шариковые и роликовые упорные подшипники при малых размерах обладают весьма высокой нагрузочной способностью, но воспринимают только осевые нагрузки и не допускают перекоса вала.

Особую группу образуют шариковые и роликовые **радиальные сферические подшипники**. Наружное кольцо такого подшипника имеет сферическую рабочую поверхность, с которой контактируют шарики или бочкообразные ролики. Сферические подшипники способны работать при перекосах вала до $2...3^\circ$ и обеспечивать его вращение в случае установки опор в раздельных корпусах.

По габаритам подшипники разделяют на семь серий диаметров и ширин: **сверхлегкую, особо легкую, легкую широкую, среднюю, среднюю широкую, тяжелую**. Однотипные подшипники с одинаковыми диаметрами внутреннего кольца, но относящиеся к разным сериям, имеют различные размеры тел качения, и, как следствие, различную нагрузочную способность.

По классам точности различают подшипники: класса 0 (**нормальной точности**); 6 (**повышенной**); 5 (**высокой**); 4 (**особо высокой**); 2 (**сверхвысокой**). От точности существенно зависит работоспособность подшипника, но одновременно с повышением точности возрастает и стоимость. Так, с переходом от класса 0 к классу 2 относительная стоимость подшипника повышается примерно в 10 раз. Поэтому использовать высокоточные подшипники в преобразователях движения ММ и роботов рекомендуется в **обоснованных случаях**.

3.3. Методика расчета подшипников качения

3.3.1. Исходные данные для расчета

В результате расчета передач и конструирования валов (см. раздел 4) редуктора определены следующие данные, используемые как исходные при расчете подшипников:

- расчетные схемы валов с внешними нагрузками и реакциями опор;
- диаметры опорных поверхностей валов;
- частоты вращения валов;
- срок службы ММ $T_{сл}$, лет;
- нагрузочная диаграмма (циклограмма);
- максимальные допустимые значения кинематической погрешности

$F'_{io\max}$ и мертвого хода $J_{l\max}$, мкм.

Кроме указанных, из технического задания на редуктор могут следовать дополнительные требования, например, максимальные допустимые габаритные размеры ММ, наличие и кратность динамических нагрузок и т.д.

3.3.2. Предварительный выбор подшипников

Подшипники предварительно назначают по **виду передачи, функции вала** в редукторе и **диаметру опорных поверхностей d_p** .

Валы цилиндрических зубчатых передач, как правило, устанавливают на шариковых радиальных подшипниках. Для конических и червячных передач необходимы радиально-упорные подшипники; в большинстве случаев в преобразователях движения ММ применяют роликовые радиально-упорные конические подшипники. В кулачковых генераторах волновых передач используют гибкие шариковые подшипники. В тех случаях, когда на вал действуют значительные осевые силы, в подшипниковые узлы включают упорные подшипники.

Для установки быстроходного и промежуточного валов трехосного цилиндрического редуктора могут быть рекомендованы подшипники средней

серии, тихоходного вала – легкой. В соосном цилиндрическом редукторе, как правило, опоры всех валов требуют подшипников средней или тяжелой серий. Обычно в обеих опорах вала располагают одинаковые подшипники.

Для валов червяков и шестерен конических передач целесообразно предварительно назначить роликовые радиально-упорные конические подшипники средней серии. Следует иметь в виду, что в передачах с постоянным направлением осевой силы менее нагруженный подшипник часто принимают более легкой серии. Опоры валов червячных колес в большинстве случаев ставят на подшипники легкой серии.

По приведенным выше рекомендациям и значению диаметра $d_{\text{п}}$ выбирают подшипники и выписывают из соответствующего стандарта их основные характеристики.

Предварительно принятые подшипники требуют проверки. Поскольку в подавляющем большинстве преобразователей движения ММ валы вращаются с частотами более 1 об/мин, то далее рассматривается проверка **по динамической грузоподъемности**.

3.3.3. Проверочный расчет подшипников по динамической грузоподъемности

3.3.3.1. Шариковые радиальные подшипники

Вал установлен в двух опорах – A и B , радиальные реакции опор – соответственно R_A и R_B , осевая реакция опоры A – R_{Aa} .

Определяют отношения

$$\frac{R_{Aa}}{VR_A}; \quad (3.1)$$

$$\frac{R_{Aa}}{C_{0r}}, \quad (3.2)$$

где V – коэффициент вращения ($V = 1$ – при вращающемся внутреннем кольце; $V = 1,2$ – при вращающемся наружном кольце подшипника); C_{0r} – табличная статическая грузоподъемность подшипника, Н.

По значению (3.2) и табл. 3.1 устанавливают коэффициент влияния осевого нагружения e .

Таблица 3.1

Коэффициенты e и Y для шариковых радиальных однорядных подшипников

R_{Aa}/C_{0r}	0,014	0,028	0,056	0,084	0,110	0,170	0,280	0,420	0,560
e	0,19	0,22	0,26	0,28	0,30	0,34	0,38	0,42	0,44
Y	2,30	1,99	1,71	1,55	1,45	1,31	1,15	1,04	1,00

Сравнение величин (3.1) и e показывает **значимость осевой силы**. Если $\frac{R_{Aa}}{VR_A} \leq e$, то в дальнейшем осевую составляющую реакции опоры A не учитывают и считают подшипник нагруженным только радиально.

Следующим шагом определяют **эквивалентные нагрузки** на подшипники.

Простейшим случаем нагружения преобразователя движения является **постоянный режим**, при котором величины реакций опор не изменяются со временем. Пусть в опоре A учет осевой реакции необходим. Тогда эквивалентные нагрузки на подшипники равны

$$\begin{aligned} P_{\text{э}A} &= (VXR_A + YR_{Aa})K_6K_T; \\ P_{\text{э}B} &= VR_BK_6K_T, \end{aligned} \quad (3.3)$$

где $X = 0,56$ – коэффициент радиальной нагрузки; Y – коэффициент осевой нагрузки (см. табл. 3.1); K_6 – коэффициент безопасности (табл. 3.2); K_T – температурный коэффициент (при рабочей температуре подшипника до 125°C $K_T = 1$).

Если же осевую реакцию учитывать не нужно, то эквивалентную нагрузку на подшипник A определяют из выражения

$$P_{\text{э}A} = VR_AK_6K_T. \quad (3.4)$$

Преобразователи движения ММ и роботов часто предназначены для выполнения вполне определенных операций, характеризующихся переменными нагрузками, скоростями движения рабочего органа и продолжительностью. В таких случаях для минимизации массы и размеров ММ необходимо учитывать **переменный характер режима нагружения**.

Учет режима нагружения производится при вычислении **приведенной эквивалентной нагрузки** на подшипник за цикл $P_{\text{эЦ}}$.

Пусть в течение цикла длительностью $t_{\text{Ц}}$ эквивалентная нагрузка на подшипник изменяется линейно от $P_{\text{эmin}}$ до $P_{\text{эmax}}$, а частота вращения вала $n = \text{const}$. Тогда приведенная эквивалентная нагрузка равна

$$P_{\text{эЦ}} = (P_{\text{эmin}} + 2P_{\text{эmax}})/3. \quad (3.5)$$

В более сложном случае, когда цикл состоит из k участков с различной нагрузкой на подшипник, приведенная эквивалентная нагрузка равна

$$P_{\text{эЦ}} = \sqrt[3]{\frac{\sum_{i=1}^k (P_{\text{э}i}^3 t_i n)}{t_{\text{Ц}}}}, \quad (3.6)$$

где $P_{эi}$ – эквивалентная нагрузка на i -ом участке; t_i – длительность i -ого участка.

В самом общем случае, когда в зависимости от участка циклограммы изменяется не только нагрузка, но и частота вращения вала, приведенная эквивалентная нагрузка равна

$$P_{эЦ} = \sqrt[3]{\frac{\sum_{i=1}^k (P_{эi}^3 t_i n_i)}{\sum_{i=1}^k (t_i n_i)}}, \quad (3.7)$$

где n_i – частота вращения вала на i -ом участке циклограммы.

Затем находят **требуемую динамическую грузоподъемность подшипников** по формулам:

для постоянного режима нагружения

$$\begin{aligned} C_{грA} &= P_{эA} \sqrt[3]{L \cdot 10^{-6}}; \\ C_{грB} &= P_{эB} \sqrt[3]{L \cdot 10^{-6}}; \end{aligned} \quad (3.8)$$

для переменного режима нагружения

$$\begin{aligned} C_{грA} &= P_{эЦA} \sqrt[3]{L \cdot 10^{-6}}; \\ C_{грB} &= P_{эЦB} \sqrt[3]{L \cdot 10^{-6}}, \end{aligned} \quad (3.9)$$

где L – число оборотов вала за срок службы ММ.

Вид выражения для вычисления L зависит от того, как в техническом задании указан срок службы ММ. Если оговорен календарный срок $T_{сл}$, лет, то число оборотов вала за $T_{сл}$ вычисляют следующим образом: для первого и второго из рассмотренных случаев

$$L = 60 T_{сл} n_{рд} n_{см} t_{см} n; \quad (3.10)$$

для третьего случая

$$L = 60 T_{сл} n_{рд} n_{см} t_{см} \sum_{i=1}^k (t_i n_i / t_{Ц}), \quad (3.11)$$

где $n_{рд}$ – число рабочих дней в году; $n_{см}$ – число рабочих смен в сутки; $t_{см}$ – число рабочих часов в смену.

В (3.10) и (3.11) $t_{Ц}$ и t_i – в минутах.

Если же задан фонд рабочего времени $T_{раб}$, то (3.10) и (3.11) соответственно преобразуются в формулы

$$L = 60T_{\text{раб}} n; \quad (3.12)$$

$$L = 60T_{\text{раб}} \sum_{i=1}^k (t_i n_i / t_{\text{Ц}}). \quad (3.13)$$

Затем сравнивают бóльшую из требуемых динамических грузоподъемностей подшипников с табличной динамической грузоподъемностью $C_{\text{ртабл}}$ предварительно принятого подшипника. Подшипник проработает заданный срок, если $C_{\text{ртр}} \leq C_{\text{ртабл}}$.

Особо следует рассмотреть проверку подшипников генераторов волн волновой передачи.

В процессе расчета передачи (см. п. 2.7.5) в соответствии с величиной делительного диаметра d_1 гибкого колеса был принят гибкий подшипник кулачкового генератора или подшипники шариковые радиальные дискового генератора. Требуемая динамическая грузоподъемность подшипника равна

$$C_{\text{ртр}} = 0,01P_{\text{эЦ}} \frac{\sqrt[3]{60T_{\text{раб}} n_{\text{п}}}}{K_L}, \quad (3.14)$$

где $P_{\text{эЦ}}$ – определяют по формуле (3.7) с подстановкой в нее $P_{\text{э}i}$, найденных по (3.15); $n_{\text{п}}$ – эквивалентная частота вращения подшипника генератора волн (формула (3.16)); K_L – коэффициент, принимаемый по табл. 2.30.

Величину эквивалентной нагрузки на i -ом участке находят по выражению

$$P_{\text{э}i} = 0,6T_{\text{т}i} d_1^{-1} V K_{\text{б}} K_{\text{т}}, \quad (3.15)$$

где $T_{\text{т}i}$ – крутящий момент на тихоходном валу на i -ом участке; $V = 1,2$; $K_{\text{б}}$ – принимают в зависимости от вида генератора волн ($K_{\text{б}} = 1,1$ – для кулачкового генератора; $K_{\text{б}} = 1,3$ – для дискового генератора); $K_{\text{т}}$ – принимают в зависимости от температуры подшипника ($K_{\text{т}} = 1$ – до 100°C включительно; $K_{\text{т}} = 1 \dots 1,05$ – от 100°C до 125°C ; $K_{\text{т}} = 1,05 \dots 1,4$ – от 125°C до 250°C).

Эквивалентная частота вращения подшипника равна

$$n_{\text{п}} = \sum_{i=1}^k (t_i n_{\text{п}i} / t_{\text{Ц}}), \quad (3.1)$$

а $n_{\text{п}i}$ определяют следующим образом: для кулачкового генератора (рис. 2.16) $n_{\text{п}i} = n_{1i}$; для дискового генератора (рис. 2.17) с подвижным гибким зубчатым колесом

$$n_{\text{ш}} = d_1(n_{3i} - n_{1i})/D_{\text{д}}; \quad (3.17)$$

для дискового генератора с подвижным жестким зубчатым колесом

$$n_{\text{ш}} = d_1(n_{3i} - n_{2i})/D_{\text{д}}. \quad (3.18)$$

Полученное значение требуемой динамической грузоподъемности не должно превышать расчетную грузоподъемность, которую для гибкого подшипника находят по формулам:

при диаметре шарика $d_{\text{ш}} \leq 25,4$ мм

$$C = f_c(j \cos \gamma)^{0,7} z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,8}; \quad (3.19)$$

при диаметре шарика $d_{\text{ш}} > 25,4$ мм

$$C = 3,647 f_c(j \cos \gamma)^{0,7} z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,4}, \quad (3.20)$$

где $f_c = 5 \dots 5,2$ – коэффициент динамической грузоподъемности; j – число рядов шариков в подшипнике (обычно $j = 1$); γ – угол контакта тел качения с кольцами; $z_{\text{ш}}$ – количество шариков в ряду.

Стандартные $z_{\text{ш}}$ и $d_{\text{ш}}$ см. в табл. 2.27.

Осевая нагрузка на гибкий подшипник равна 0,1 радиальной, поэтому

$$\gamma = \arctg 0,1 = 5,71^\circ = 5^\circ 43', \quad (3.21)$$

и формулы (3.19) и (3.20) для $j = 1$ принимают вид

$$C = 0,996 f_c z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,8} \approx f_c z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,8}; \quad (3.22)$$

$$C = 3,63 f_c z_{\text{ш}}^{0,67} d_{\text{ш}}^{1,4}. \quad (3.23)$$

Расчетные динамические грузоподъемности шариковых радиальных подшипников дисковых генераторов волн указаны в стандарте на подшипники.

При выполнении условия $C_{\text{тр}} \leq C$ принятый подшипник будет обладать заданной долговечностью $T_{\text{раб}}$.

Может быть также использован другой способ оценки долговечности гибкого подшипника.

Определяют эквивалентный крутящий момент на тихоходном валу за рабочий цикл

$$T_{\text{э}} = \sum_{i=1}^k (T_{\text{ти}} t_i n_{\text{ти}} / t_{\text{ц}}), \quad (3.24)$$

после чего находят расчетную долговечность подшипника по формуле

$$L_h = 10^4 (n_{\text{max}} / n_{\text{п}}) (T_{\text{max}} / T_{\text{э}}), \quad \text{ч}, \quad (3.25)$$

где n_{max} – предельная частота вращения для подшипника по табл. 2.27; T_{max} – допустимый крутящий момент на тихоходном валу (табл. 3.2).

3.3.3.2. Радиально-упорные подшипники

В радиально-упорных подшипниках контактные линии наклонены к оси вала на угол α , что при радиальном нагружении приводит к появлению **внутренних осевых сил** S (рис. 3.1).

Значения S определяют так:
для шариковых подшипников

$$S = Re; \quad (3.26)$$

для роликовых подшипников

$$S = 0,83Re, \quad (3.27)$$

где e – параметр осевой нагрузки (по каталогу).

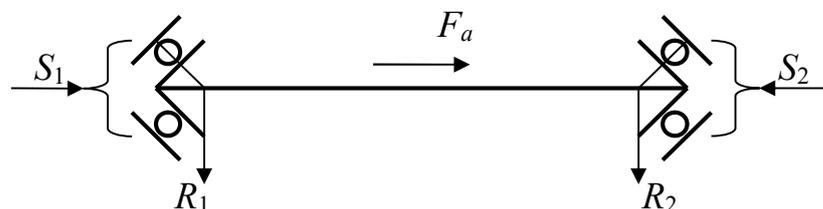


Рис. 3.1. Схема вала на радиально-упорных подшипниках (F_a – внешняя осевая сила)

Осевые нагрузки на подшипники вычисляют по формулам, приведенным в табл. 3.3.

Таблица 3.2

Допустимый крутящий момент на тихоходном валу волновой передачи T_{\max} , Нм

Диаметр наружного кольца D , мм	Передаточное число u	Допустимый крутящий момент T_{\max}
100	80	180
	100	200
	125	224
	160	250
	200 и более	280
120	80	355
	100	400
	125	450
	160	500
	200 и более	560
160	80	710
	100	800
	125	900
	160 и более	1000
200	80	1400
	100	1600
	125	1800
	160 и более	2000
240	80	2800
	100	3150
	125 и более	3550
320	80	5600
	100 и более	6300

Таблица 3.3

Расчет осевых нагрузок на подшипники вала по рис. 3.1

Условия нагружения	Осевые нагрузки
$S_1 > S_2; F_a > 0$	$F_{a1} = S_1;$
$S_1 < S_2; F_a > S_2 - S_1$	$F_{a2} = S_1 + F_a$
$S_1 < S_2; F_a < S_2 - S_1$	$F_{a1} = S_2 - F_a; F_{a2} = S_2$

Эквивалентную динамическую нагрузку на роликовый подшипник следует определять по формулам:

$$\begin{aligned} C_{r\text{тp}} &= P_{\text{э}} \sqrt[3,33]{L \cdot 10^{-6}}; \\ C_{r\text{тp}} &= P_{\text{эЦ}} \sqrt[3,33]{L \cdot 10^{-6}}. \end{aligned} \tag{3.28}$$

В остальном методика проверки аналогична приведенной в подпункте 3.3.3.1.

4. ВАЛЫ И ОСИ

4.1. Общие сведения

Валы и оси служат для размещения на них вращающихся деталей: зубчатых колес, шкивов, барабанов и т.п. Отличие вала от оси состоит в том, что он передает крутящий момент от одной детали к другой, а ось не передает. **Вал** всегда **вращается**, а **ось** может быть как **вращающейся**, так и **не вращающейся**.

Различают валы **прямые**, **коленчатые** и **гибкие**. В преобразователях движения ММ и роботов, как правило, применяются прямые валы. Коленчатые и гибкие валы относятся к специальным деталям и далее рассматриваться не будут.

По наличию ступеней валы делятся на **гладкие** и **ступенчатые**. Наличие ступеней связано с установкой на валу деталей и самого вала в опорах. В некоторых случаях выполнение вала или оси ступенчатой формы позволяет существенно уменьшить их массу.

Валы изготавливают **сплошными** или **полыми**. Полость либо уменьшает массу вала, либо позволяет пропустить через вал другую деталь, подвести масло к контактирующим поверхностям и т.п.

Для изготовления валов и осей применяют преимущественно **углеродистые** и **легированные стали**, предусматривающие все возможные виды упрочнения.

Далее приводится методика расчета вала преобразователя движения ММ.

4.2. Методика расчета валов

4.2.1. Исходные данные

Размеры устанавливаемых на вал элементов (зубчатых и червячных колес, посадочные диаметры подшипников и т.п.).

Значения нагрузок на эти элементы. Передаваемый валом крутящий момент.

Частота вращения вала.

Циклограмма нагружения и срок службы ММ.

Дополнительные требования (ориентировочные габаритные размеры, тип подшипников и пр.).

4.2.2. Проектировочный расчет

Определяют **характерный диаметр** вала

$$d = 10^3 \sqrt[3]{\frac{T}{0,2[\tau]}}, \quad (4.1)$$

где T – передаваемый валом крутящий момент, Нм; $[\tau] = 20$ МПа – уменьшенное допускаемое касательное напряжение.

Для быстроходных и тихоходных валов d – диаметр выходного конца, его значение следует принять ближайшее по ГОСТ 12080 или ГОСТ 12081 к полученному по (4.1).

Для промежуточных валов d – диаметр ступеньки под зубчатым (или червячным) колесом, его значение следует принять ближайшее по ряду $Ra40$ к полученному по (4.1).

Диаметры остальных ступенек вала назначают с учетом величины характерного диаметра конструктивно, принимая во внимание требования технологии изготовления и сборки, ряд номинальных размеров, вероятные размеры подшипников и известные размеры сопряженных с валом деталей.

4.2.3. Разработка расчетной схемы

Расчетная схема представляет собой схематичное изображение вала в виде двухопорной балки. Нагрузки показывают в виде сосредоточенных сил и моментов. При необходимости балку снабжают пометками с указанием диаметров ступенек.

Быстроходный вал преобразователя движения ММ соединяют с валом двигателя без компенсирующей муфты, поэтому консольная нагрузка на его выходной конец отсутствует. Часто шестерню быстроходной ступени редуктора насаживают прямо на вал двигателя, в результате чего необходимость в быстроходном вале отпадает.

На выходной конец тихоходного вала преобразователя движения ММ общего назначения может действовать **консольная нагрузка**, которую учитывают в виде силы, определяемой по следующим формулам: для одноступенчатого цилиндрического редуктора

$$F_k = 125\sqrt{T_t}, \text{ Н}; \quad (4.2)$$

для двухступенчатого цилиндрического редуктора и для червячного редуктора

$$F_k = 250\sqrt{T_t}, \text{ Н}, \quad (4.3)$$

где T_t – в ньютонметрах.

Длины ступенек при разработке расчетной схемы назначают с учетом размеров деталей, ряда номинальных размеров, а также соотношений, принятых в практике конструирования.

4.2.4. Построение эпюр изгибающих и крутящих моментов в сечениях вала

Данный пункт расчета подробно рассматривается в сопротивлении материалов.

4.2.5. Проверка вала на усталостную прочность

Методика проверки достаточно полно изложена в [7]. Здесь же остановимся на нижеследующем отличии проверки вала преобразователя движения ММ от проверки вала редуктора общего назначения.

Разрабатывая преобразователь движения ММ, техническое задание на который содержит требование **обеспечения минимальной массы**, конструктор обязан решить вопрос о возможности и целесообразности корректировки диаметров ступенек вала по результатам проверки.

Пусть в опасном сечении вала диаметром d_{oc} запас усталостной прочности больше допустимого, т.е. $s > [s]$. Тогда скорректированное значение диаметра достаточно точно может быть определено по формуле

$$d'_{oc} \approx d_{oc} \sqrt[3]{\frac{[s]}{s}}. \quad (4.4)$$

Допускаемый запас прочности, рекомендуемый в [7], $[s] = 1,6 \dots 2,1$, но при повышенных требованиях к жесткости следует принимать $[s] = 2,5 \dots 3$ [8]. В случае, когда к ММ предъявляются высокие требования в части точности, желательно обеспечить $[s] = 2,5$. В силу различных причин (минимизация массогабаритных показателей, определенная компоновка преобразователя движения и пр.) $[s]$ может быть понижен до $[s] = 1,6$, но тогда появляется опасность возникновения слишком большой погрешности в передаче из-за деформации вала.

Изменение диаметра одного участка при сохранении принятого соотношения диаметров повлечет за собой соответствующее уменьшение диаметров остальных участков, в том числе – под подшипниками. На предыдущем этапе (п. 3.3) принятые предварительно подшипники были проверены по динамической грузоподъемности, их серия, а, возможно, и тип были подобраны таким образом, чтобы максимально приблизить $C_{rтр}$ к C_r . Корректировка же диаметров вала делает эти подшипники непригодными, поэтому следует заново назначить подшипники и выполнить их проверку.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Иванов М.П., Финогенов В.А. Детали машин. – М.: Высш. шк., 2003.
2. Анурьев В.И. Справочник конструктора-машиностроителя: В 3-х т. Т3. – М.: Машиностроение, 1992.
3. Таугер В.М. Конструирование преобразователей движения мехатронных модулей: Учеб. пособие. – Екатеринбург: УрГУПС, 2006.
4. Таугер В.М., Ахлюстина Н.В. Расчет и курсовое проектирование деталей машин: В 2-х ч. Ч.1. – Екатеринбург: УрГУПС, 2004.
5. Анурьев В.И. Справочник конструктора-машиностроителя: В 3-х т. Т2. – М.: Машиностроение, 1992.
6. Анурьев В.И. Справочник конструктора-машиностроителя: В 3-х т. Т1. – М.: Машиностроение, 1992.
7. Дунаев П.Ф., Леликов О.П. Детали машин. Курсовое проектирование. – М.: Машиностроение, 2004.
8. Проектирование механических передач: Учеб.-справ. пособие для вузов/С.А. Чернавский, Г.А. Снесарев, Б.С. Козинцов и др. – М.: Машиностроение, 1984.
9. Красковский Е.Я., Дружинин Ю.А., Филатова Е.М. Расчет и конструирование механизмов приборов и вычислительных систем: Учеб. пособие для студентов вузов. – М.: Высш. шк., 1983.
10. Егоров О.Д. Точность манипуляционных механизмов роботов. – М.: Изд-во МПИ, 1989.

И. о. проректора по учебно-методической
работе В. В. Зубов



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Б1.В.09.05 МЕЖДУНАРОДНЫЙ ИНЖИНИРИНГ

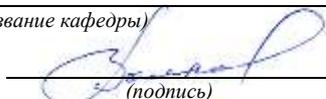
*Специальность –
21.05.04 Горное дело*

*Специализация –
Мехатроника и робототехника промышленных комплексов*

Одобрена на заседании кафедры

Технической механики
(название кафедры)

Зав. кафедрой


(подпись)

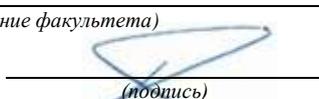
Волков Е.Б.
(Фамилия И.О.)

Протокол № 1 от 13.09.2024
(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического
(название факультета)

Председатель


(подпись)

Осипов П.А.
(Фамилия И.О.)

Протокол № 2 от 18.10.2024
(Дата)

Екатеринбург

СОДЕРЖАНИЕ

Введение.....	4
Методические рекомендации для самостоятельной работы	5
Методические рекомендации для практических занятий.....	7
Методические рекомендации для выполнения контрольной работы..	11
Библиографический список.....	14

ВВЕДЕНИЕ

Общей тенденцией развития науки и техники последнего столетия является выступая интеграция наук, преодоление принципа декомпозиции, стремление к системному подходу. Свидетельством этому является возникновение новых научно-технических направлений, таких, как кибернетика, бионика, системотехника и мехатроника, которая базируется на системных знаниях в областях механики, электроники и компьютерного управления функциональными движениями.

Именно концепция мехатроники наиболее полно отражает глобальную тенденцию интеграции всех функциональных компонентов технических систем вплоть до их конструктивного слияния в виде единых конструкций современных автоматических и автоматизированных систем и комплексов на основе решения двух, тоже общих, проблем развития техники в целом – миниатюризации и интеллектуализации.

Мехатроника является новой и динамично развивающейся отраслью науки и техники. Она базируется на знаниях и достижениях в областях механики, электроники и компьютерного управления и представляет собой более высокий уровень развития современного машиностроения.

Основной целью дисциплины «Международный инжиниринг» является подготовка студентов к научно-исследовательской деятельности посредством обеспечения этапов формирования соответствующих компетенций. Наряду с практической целью курс ставит образовательные и воспитательные цели. Достижение этих целей означает расширение кругозора студентов, повышение уровня их общей культуры и образования, а также культуры мышления, общения и речи и проявляется в готовности студента содействовать налаживанию межкультурных, профессиональных связей.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа – это вид учебной деятельности, выполняемый студентом без непосредственного контакта с преподавателем или управляемый преподавателем опосредовано через специальные учебные материалы; неотъемлемое обязательное звено процесса обучения, предусматривающее, прежде всего, индивидуальную работу студентов в соответствии с установкой преподавателя или учебника, программы обучения.

В процессе самостоятельной деятельности студент должен научиться выделять познавательные задачи, выбирать способы их решения, выполнять операции контроля за правильностью решения поставленной задачи, совершенствовать навыки реализации теоретических знаний.

Самостоятельная работа студента под руководством преподавателя протекает в форме делового взаимодействия: студент получает непосредственные указания, рекомендации преподавателя об организации самостоятельной деятельности, а преподаватель выполняет функцию управления через учет, контроль и коррекцию ошибочных действий.

Успешность самостоятельной работы в первую очередь определяется степенью подготовленности студента. По своей сути самостоятельная работа предполагает максимальную активность студентов в различных аспектах: организации умственного труда, поиске информации, стремлении сделать знания убеждениями. Психологические предпосылки развития самостоятельности студентов заключаются в их успехах в учебе, положительном к ней отношении, заинтересованности и увлеченности предметом, помимо того, что при правильной организации самостоятельной работы приобретаются навыки и опыт творческой деятельности.

Самостоятельная работа обучающихся является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих систем, а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачетам и экзаменам.

Самостоятельная работа предназначена не только для овладения дисциплиной, но и для формирования навыков самостоятельной работы вообще, в учебной, научной, профессиональной деятельности, способности принимать на себя ответственность, самостоятельно решить проблему, находить конструктивные решения, выход из кризисной ситуации и т. д.

Самостоятельная работа завершает задачи всех видов учебной работы. Никакие знания, не подкрепленные самостоятельной деятельностью, не могут стать подлинным достоянием человека. Кроме того, самостоятельная работа имеет воспитательное значение: она формирует самостоятельность не только как совокупность умений и навыков, но и как черту характера, играющую существенную роль в структуре личности современного специалиста высшей квалификации.

Для повышения эффективности самостоятельной работы студентов используется:

- организация индивидуальных планов обучения с привлечением студентов к научно-исследовательской работе и по возможности к реальному проектированию по заказам предприятий;
- включение самостоятельной работы студентов в учебный план и расписание занятий с организацией индивидуальных консультаций на кафедрах;
- создание комплекса учебных и учебно-методических пособий для выполнения самостоятельной работы студентов;
- разработка системы интегрированных межкафедральных заданий;
- ориентация лекционных курсов на самостоятельную работу;
- рейтинговый метод контроля самостоятельной работы студентов;
- коллегиальные отношения преподавателей и студентов;
- разработка заданий, предполагающих нестандартные решения;
- индивидуальные консультации преподавателя и перерасчет его учебной нагрузки с учетом самостоятельной работы студентов;
- проведение форм лекционных занятий типа лекции-беседы, лекции-дискуссии, где докладчиками и содокладчиками выступают сами студенты, а преподаватель выполняет роль ведущего. Такие занятия предполагают предварительную самостоятельную проработку каждой конкретной темы выступающими студентами по учебным пособиям, консультации с преподавателем и использование дополнительной литературы.

Тема 1. Materials technology. Describing and categorizing materials. Specifying and describing properties. Discussing quality issue. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - Present Simple, Present Continuous. Nouns and Pronouns. Numerals.

Тема 2. Component shapes and features. Manufacturing techniques. Jointing and fixing techniques. Positions of assembled components. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - irregular verbs; Past Simple vs. Present Perfect; articles.

Тема 3. Engineering design. Technical drawing (CAD, CAM). Design phases and procedures. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение

письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - Speaking about future, adjectives degrees of comparisons.

Тема 4. Electrical circuit concept. Electrical generation and transmission. Electrical loads. Semiconductor devices. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - other perfect tenses, passive and active voice.

Тема 5. Signal processing. Power electronics. Electromagnetisms. Electrica effect and devices. Sensors and actuators. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - Modals.

Тема 6. Digital logic. Control systems - models, operation principles. Software engineering. Describing automated systems. Robotics. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - conditionals; Gerund and Infinitive.

Тема 7. International patenting practice. Norms and regulations in this sphere. Structural, stylistics and linguistic specificity of documentation used in the international patenting practice. Patent search and analysis. Writing a patent. General grammar rules revision and practice.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Практические занятия на уроках дисциплины «Международный инжиниринг» направлены на закрепление теоретических знаний и формирование профессиональных практических умений. Проведение практических занятий позволяет заинтересовать обучающихся в изучении предмета, способствует активному усвоению знаний и умений сбора, обработки и анализа информации, характеризующей различные ситуации. Практические занятия развивают такие профессионально значимые качества, как самостоятельность, ответственность, точность, творческую инициативу, исследовательские умения (наблюдать,

сравнивать, анализировать, устанавливать зависимость, делать выводы и обобщения). Содержание разработанных практических занятий направлено на реализацию Государственных требований и требований работодателя.

Выполнению практических заданий предшествует проверка знаний обучающихся – их теоретической готовности к выполнению данных заданий. К каждому практическому занятию разработана инструкция для обучающихся, в которой указан порядок необходимых действий, а также контрольные вопросы. Основная позиция обучаемого в учебном процессе – активно-деятельностная, субъектная – включает в себя самостоятельный поиск, принятие решений, оценочную деятельность. Основная позиция преподавателя – руководитель и партнер по выполнению практических заданий.

С целью контроля и подготовки студентов к изучению новой темы в начале каждого практического занятия преподавателем проводится индивидуальный или фронтальный устный опрос по выполненным заданиям предыдущей темы. Критерии оценки: – правильность ответа по содержанию задания (учитывается количество и характер ошибок при ответе);

– полнота и глубина ответа (учитывается количество усвоенных фактов, понятий и т.п.);

– владение словарем по тематике задания (количество освоенных новых слов);

– сознательность ответа (учитывается понимание излагаемого материала);

– логика изложения материала (учитывается умение строить целостный, последовательный рассказ, грамотно пользоваться специальной терминологией);

– рациональность использованных приемов и способов решения поставленной учебной задачи (учитывается умение использовать наиболее прогрессивные и эффективные способы достижения цели);

– своевременность и эффективность использования наглядных пособий и технических средств при ответе (учитывается грамотно и с пользой применять наглядность и демонстрационный опыт при устном ответе);

– использование дополнительного материала (обязательное условие);

– рациональность использования времени, отведенного на задание (не одобряется затянутость выполнения задания, устного ответа во времени, с учетом индивидуальных особенностей студентов).

Критерии оценки устных ответов студентов

Оценочные средства текущего контроля успеваемости: тест, опрос.

Система оценивания по оценочным средствам текущего контроля

Оценочное средство	Балловая стоимость
Тест	0-50 баллов (10 заданий)
Опрос	0-10 баллов
Итого	60 баллов

Баллы за тестирование проставляются за правильные ответы на вопросы.

В зависимости от типа вопроса ответ считается правильным, если:

- в тестовом задании закрытой формы с выбором ответа выбран правильный ответ;
- в тестовом задании открытой формы дан правильный ответ;
- в тестовом задании на установление правильной последовательности установлена правильная последовательность;
- в тестовом задании на установление соответствия, если сопоставление произведено верно для всех пар.

Оценка за опрос определяется простым суммированием баллов:

Критерии оценки ответа на вопрос	Количество баллов
правильность ответа	4
всесторонность и глубина ответа (полнота)	3
наличие выводов	1
соблюдение норм литературной речи	1
владение профессиональной лексикой	1
Итого	10

Результаты текущего контроля фиксируются преподавателем.

Типовые контрольные задания и материалы

Тест:

1. К какому виду услуг относится оценка стоимости проекта?

- А) Инжиниринговые услуги;
- Б) Финансовые услуги;
- В) Консультативные услуги;
- Г) Правильного ответа нет.

2. Как называется оказание услуг производственного, коммерческого, инженерно-проектного и научно-технического характера в международном масштабе?

- А) Межнациональные договора;
- Б) Международный рейтинг;
- В) Международный инжиниринг;
- Г) Нет нужного ответа.

3. Кем может быть оказан международный инжиниринг?

- А) Поставщиком технологии;
- Б) Генеральным подрядчиком;
- В) Производителем;
- Г) Правильного ответа нет.

4. Что представляют собой инжиниринговые услуги?

- А) Услуги, связанные с подготовкой и налаживанием производственного процесса;
- Б) Услуги, связанные с оказанием консультационных услуг;
- В) Услуги, связанные с осуществлением предпринимательской деятельности;
- Г) Правильного варианта нет.

Методические материалы, определяющие процедуры оценивания знаний, умений, владений (опыта деятельности), характеризующие формирование компетенций

1. Ознакомление обучающихся с процедурой и алгоритмом оценивания (в начале изучения дисциплины).

2. Проверка ответов на задания, выполненных работ.

3. Сообщение результатов оценивания обучающимся, обсуждение результатов.
4. Оформление необходимой документации.

Перечень вопросов для устного опроса

Materials technology. Describing and categorizing materials.
Specifying and describing properties of materials.
Component shapes and features.
Manufacturing techniques.
Jointing and fixing techniques.
Positions of assembled components.
Engineering design.
Technical drawing (CAD, CAM). Design phases and procedures.
Manufacturing techniques.
Jointing and fixing techniques.
Electrical circuit concept.
Electrical generation and transmission.
Electrical loads.
Semiconductor devices.
Signal processing.
Power electronics.
Electromagnetisms.
Electrical effect and devices.
Sensors and actuators.
Digital logic.

Практические занятия

1. Materials technology. Describing and categorizing materials. Specifying and describing properties. Discussing quality issue. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - Present Simple, Present Continuous. Nouns and Pronouns. Numerals.
2. Engineering design. Technical drawing (CAD, CAM). Design phases and procedures. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - Speaking about future, adjectives degrees of comparisons.
3. Component shapes and features. Manufacturing techniques. Jointing and fixing techniques. Positions of assembled components. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - irregular verbs; Past Simple vs. Present Perfect; articles.
4. Electrical circuit concept. Electrical generation and transmission. Electrical loads. Semiconductor devices. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - other perfect tenses, passive and active voice.
5. Signal processing. Power electronics. Electromagnetisms. Electrical effect and devices. Sensors and actuators. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройденным темам. повторение и систематизация ранее пройденного грамматического материала - Modals.
6. Digital logic. Control systems - models, operation principles. Software engineering. Describing automated systems. Robotics. Выполнение лексико-грамматических упражнений, чтение текста и соответствующие предтекстовые и послетекстовые задания. Выполнение письменного задания, монологическое и диалогические высказывания по пройден-

ным темам. повторение и систематизация ранее пройденного грамматического материала - conditionals; Gerund and Infinitive.

7. International patenting practice. Norms and regulations in this sphere. Structural, stylistics and linguistic specificity of documentation used in the international patenting practice. Patent search and analysis. Writing a patent. General grammar rules revision and practice.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ

Контрольная работа – это одна из форм устной итоговой аттестации, это самостоятельная исследовательская работа, в которой автор раскрывает суть исследуемой проблемы; приводит различные точки зрения, а также собственные взгляды на нее. Содержание работы должно быть логичным; изложение материала носит проблемно-тематический характер.

Контрольная работа, как форма текущей аттестации, стимулирует раскрытие исследовательского потенциала студента, способность к творческому поиску, сотрудничеству, самораскрытию и проявлению возможностей.

Автор работы должен продемонстрировать достижение им уровня формируемых компетенций, продемонстрировать знание предмета, умение проявлять оценочные знания, изучать теоретические работы, использовать различные методы исследования, применять различные приемы творческой деятельности. Для этого необходимо правильно сформулировать тему, отобрать по ней необходимый материал; использовать только тот материал, который отражает сущность темы. Изложение должно быть последовательным. Недопустимы нечеткие формулировки, речевые и орфографические ошибки. В подготовке работы необходимо использовать материалы современных изданий. Оформление реферата должно быть грамотным и соответствовать ГОСТ 2.105–95, ГОСТ 9327-60. Библиографический список оформляется в соответствии с ГОСТ 7.1-2003.

Изложение текста и оформление выполняют в соответствии с требованиями ГОСТ. Страницы текстовой части и включенные в нее иллюстрации и таблицы должны соответствовать формату А4 по ГОСТ 9327-60. Работа должна быть выполнена печатным способом на одной стороне бумаги формата А4 через полтора интервала. Цвет шрифта должен быть черным, высота букв, цифр и других знаков не менее 1.8 (шрифт Times New Roman, 14 пт.). Текст следует печатать, соблюдая следующие размеры полей: верхнее и нижнее — 20 мм, левое — 30 мм, правое — 10 мм. Абзацный отступ должен быть одинаковым по всему тексту и составлять 1,25 см. Выравнивание текста по ширине. Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах, формулах, применяя выделение жирным шрифтом, курсив, подчеркивание. Точку в конце заголовка не ставят. Если заголовок состоит из двух предложений, их разделяют точкой. Название каждой главы и параграфа в тексте работы можно писать более крупным шрифтом, жирным шрифтом,

чем весь остальной текст. Каждая глава начинается с новой страницы, параграфы (подразделы) располагаются друг за другом.

Все страницы обязательно должны быть пронумерованы. Нумерация листов должна быть сквозной. Номер листа проставляется арабскими цифрами. Нумерация листов начинается с третьего листа (после содержания) и заканчивается последним. Номер страницы на титульном листе не проставляется. Список использованной литературы и приложения включаются в общую нумерацию листов. Рисунки и таблицы, расположенные на отдельных листах, включают в общую нумерацию листов и помещают по возможности следом за листами, на которых приведены ссылки на эти таблицы или иллюстрации. Таблицы и иллюстрации нумеруются последовательно арабскими цифрами сквозной нумерацией. Допускается нумеровать рисунки и таблицы в пределах раздела. В этом случае номер таблицы (рисунка) состоит из номера раздела и порядкового номера таблицы, разделенных точкой. В конце работы размещаются приложения. В тексте на все приложения должны быть даны ссылки. Каждое приложение следует начинать с новой страницы с указанием наверху посередине страницы слова «Приложение» и его номера. Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

Контрольная работа оценивается по следующим критериям: актуальность темы; соответствие содержания теме; глубина проработки материала; правильность и полнота использования источников; соответствие оформления стандартом.

На «отлично»: присутствие всех вышеперечисленных требований; знание студентом изложенного материала, умение грамотно и аргументировано изложить суть проблемы; присутствие личной заинтересованности в раскрываемой теме, собственную точку зрения, аргументы и комментарии, выводы; умение свободно беседовать по любому пункту плана, отвечать на вопросы, поставленные по теме работы; умение анализировать фактический материал и статистические данные, использованные при написании работы; наличие качественно выполненного презентационного материала или (и) раздаточного, не дублирующего основной текст защитного слова, а являющегося его иллюстративным фоном.

На «хорошо»: мелкие замечания по оформлению реферата; незначительные трудности по одному из перечисленных выше требований.

На «удовлетворительно»: тема контрольной работы раскрыта недостаточно полно; неполный список литературы и источников; затруднения в изложении, аргументировании.

Контрольная работа должна содержать: титульный лист, с обязательным указанием наименования учреждения, в котором выполнялся реферат, Ф.И.О. автора, Ф.И.О. руководителя, год написания; введение; основную часть; заключительные выводы; список литературы; при необходимости приложения.

Задание для контрольной работы

1. Write a simple explanation of the existence and behavior of holes in semi-conductors.
2. Briefly describe the process by which purified silicon wafers are produced for electronics use from common sand.
3. Draw a sketch similar to Figure 6.3 for p-type Si with boron doping.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Cambridge English for Engineering / Ibbotson M., Day J. – Cambridge: Cambridge university press, 2012
2. Oxford English for Electrical and Mechanical Engineering / Answer Book with teaching notes / Glendinning E. H., Glendinning N. – Oxford : University Press, 2006
3. Technology-1: students book: oxford English for careers / Glendinning E. H., Pohl Alison. – New York: Oxford University Press, 2007



УТВЕРЖДАЮ
И. о. проректора по учебно-методической
работе _____ В. В. Зубо

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ**

Б1.В.09.06 СРЕДСТВА ИЗМЕРЕНИЙ

Специальность -
21.05.04 Горное дело

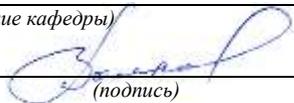
Специализация -
Мехатроника и робототехника промышленных комплексов

Одобрена на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой



(подпись)

Волков Е.Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

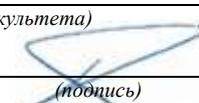
(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического

(название факультета)

Председатель



(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

СОДЕРЖАНИЕ

1 Цель работы	4
2 Задание	4
3 Необходимые приборы и пособия	4
4 Основные теоретические сведения	4
5 Методические указания по выполнению работы	13
6 Содержание отчета	16
7 Контрольные вопросы	17
Список литературы	18
Приложение А	19
Приложение Б	22

1 ЦЕЛЬ РАБОТЫ

1.1 Практическое знакомство с методами расчета сопряжений с учетом допусков и посадок по ГОСТ 25347-82.

1.2 Получение навыков практического пользования таблицами допусков и посадок.

1.3 Освоение правил графического построения полей допусков, зазоров и натягов.

2 ЗАДАНИЕ

2.1 Для заданных посадок определить наибольшие, наименьшие предельные размеры и допуски размеров деталей, входящих в соединение;

2.2 Построить схемы расположения полей допусков деталей, входящих в соединение;

2.3 Определить наибольшие, наименьшие зазоры и натяги и допуски посадок; полученные данные занести в таблицу.

3 НЕОБХОДИМЫЕ ПРИБОРЫ И ПОСОБИЯ

ГОСТ 25347-82. ЕСДП. Поля допусков и рекомендуемые посадки.

4 ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

4.1 Применяемые условные обозначения:

d – диаметр вала;

D – диаметр отверстия;

ES – верхнее отклонение отверстия;

EI – нижнее отклонение отверстия;

es – верхнее отклонение вала;

ei – нижнее отклонение вала;

d_{\max} , d_{\min} – наибольший и наименьший размеры вала;

D_{\max} , D_{\min} – наибольший и наименьший размеры отверстия;

S_{\max} , S_{\min} – наибольший и наименьший зазоры;

N_{\max} , N_{\min} – наибольший и наименьший натяги.

4.2 Основные понятия и терминология

Машины и механизмы состоят из деталей, которые в процессе работы должны совершать относительные движения или находиться в относительном покое. В большинстве случаев детали машин представляют собой определенные комбинации геометрических тел, ограниченных поверхностями простейших форм: плоскими, цилиндрическими, коническими и т. д. Это объясняется широким использованием в механизмах низших кинематических пар и технологическими соображениями, так как существующие станки приспособлены в основном для обработки простейших поверхностей и их комбинаций. Простейшие геометрические тела, составляющие детали, – будем называть их элементами.

Две детали, элементы которых входят друг в друга, образуют *соединение*. Такие детали называются *сопрягаемыми деталями*, а поверхности соединяемых элементов – *сопрягаемыми поверхностями*. Поверхности тех элементов деталей, которые не входят в соединение с поверхностями других деталей, называются *несопрягаемыми поверхностями*. Соединения подразделяются по геометрической форме сопрягаемых поверхностей. Соединение деталей, имеющих сопрягаемые цилиндрические поверхности с круглым поперечным сечением, называется *гладким цилиндрическим* (рисунок 4.1, а).

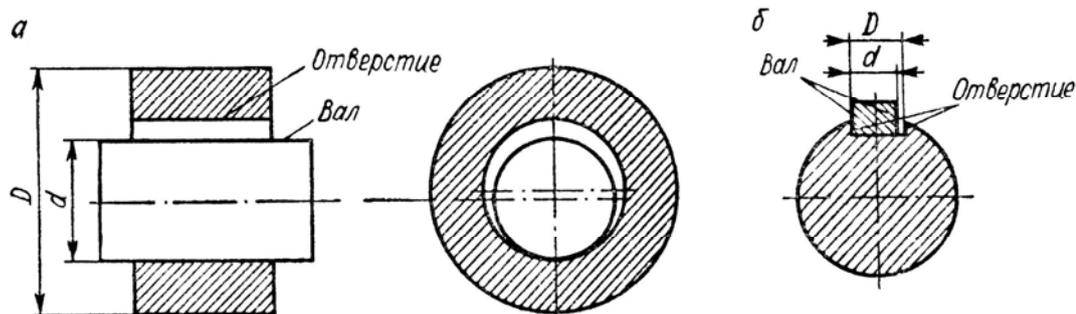


Рисунок 4.1 – Типы соединения деталей

Если сопрягаемыми поверхностями каждого элемента соединения являются две параллельные плоскости, то соединение называется *плоским соединением* с параллельными плоскостями или просто *плоским* (рис. 4.1, б).

В соединении элементов двух деталей один из них является внутренним (охватывающим), другой – наружным (охватываемым). В системе допусков и посадок гладких соединений всякий наружный элемент условно называется *валом*, всякий внутренний – *отверстием*. Термины «отверстие» и «вал» применяются и к несопрягаемым элементам.

Под *размером* элементов, образующих гладкие соединения, и аналогичных несопрягаемых элементов понимается: в цилиндрических соединениях – диаметр, в плоских – расстояние между параллельными плоскостями по нормали к ним. В более узком смысле в системе допусков и посадок *размер* – числовое значение линейной величины (диаметра, длины и т. д.) в выбранных единицах измерения (в машиностроении обычно в миллиметрах).

Номинальный размер – основной размер, полученный на основе расчетов и указанный на чертеже. Он служит началом отчета отклонений и относительно его определяются предельные размеры.

Действительный размер – размер элемента, установленный измерением, с допустимой погрешностью.

Отклонение – разность между действительным и номинальным размерами.

Предельные отклонения – два предельно допустимых размера, между которыми должен находиться действительный размер.

Верхнее отклонение – алгебраическая разность между наибольшим предельным и номинальным размерами.

ES, es – соответственно, верхнее отклонение отверстия и вала:

$$ES = D_{\max} - D; es = d_{\max} - d. \quad (4.1)$$

Нижнее отклонение – алгебраическая разность между наименьшим предельным и номинальным размерами.

EI, ei – соответственно, нижнее отклонение отверстия и вала:

$$EI = D_{\min} - D; ei = d_{\min} - d. \quad (4.2)$$

Допуск T – разность между наибольшим и наименьшим предельными размерами или алгебраическая разность между верхним и нижним отклонениями (рисунок 4.2):

$$T = D_{\max} - d_{\min}; T = d_{\max} - D_{\min}$$

или $T = ES - ei; T = es - EI.$ (4.3)

Примечание. Допуск – это абсолютная величина без знака.

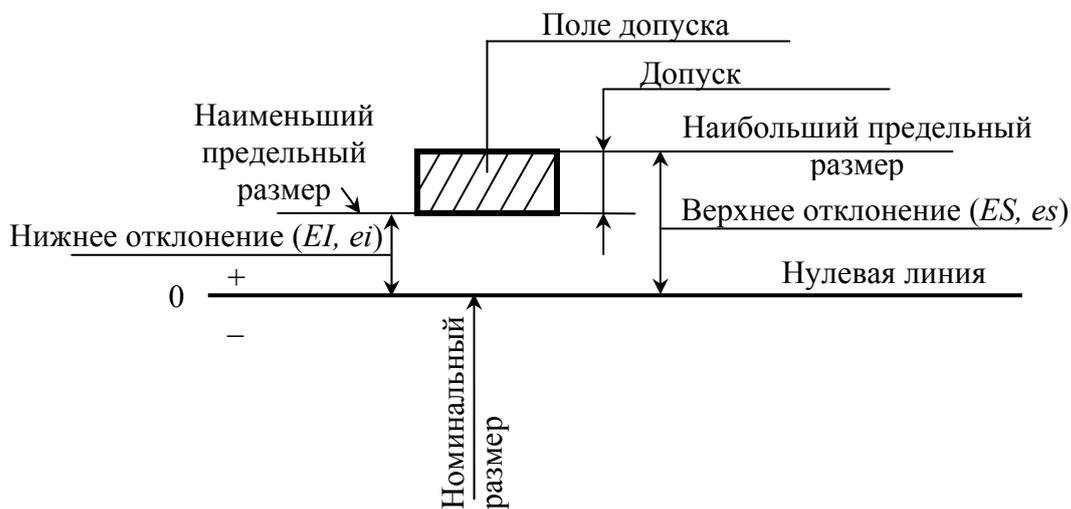


Рисунок 4.2 – Поле допуска

Поле допуска – поле между предельными отклонениями размера: оно определяется величиной допуска и его положением относительно номинального размера (рисунок 4.2).

Нулевая линия – линия, соответствующая номинальному размеру. При графическом изображении полей допусков и посадок от нее откладываются отклонения размеров (рисунок 4.2):

- положительные отклонения – вверх от нее;
- отрицательные отклонения – вниз.

Квалитет (степень точности) – совокупность допусков, которые соответствуют одному уровню точности для всех номинальных размеров.

Стандарт устанавливает 20 квалитетов: 01, 0, 1, 2, ... 18. В основном применяются квалитеты, начиная с 5-го.

Разность размеров отверстия и вала до сборки определяет характер соединения деталей, или *посадку*, т. е. большую или меньшую свободу относительного перемещения деталей или степень сопротивления их взаимному смещению. Разность размеров отверстия и вала, если размер отверстия больше размера вала, называется *зазором* S (рисунок 4.3, а):

$$S = D - d,$$

$$S_{\max} = D_{\max} - d_{\min} \text{ или } S_{\max} = ES - ei,$$

$$S_{\min} = D_{\min} - d_{\max} \text{ или } S_{\min} = EI - es. \quad (4.4)$$

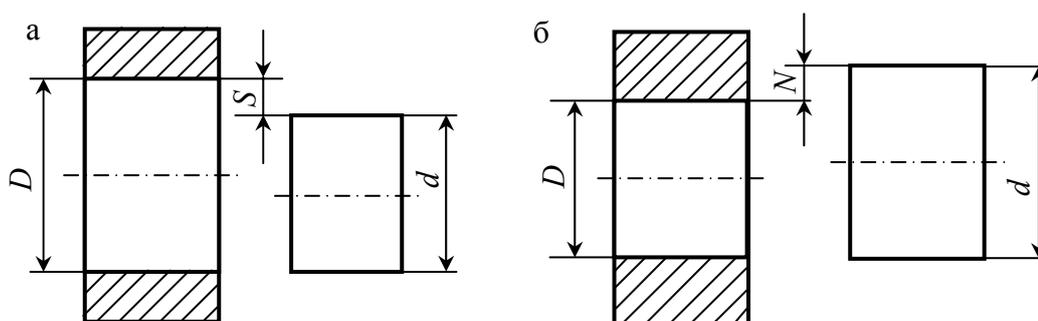


Рисунок 4.3 – Зазор и натяг

Зазор характеризует большую или меньшую свободу относительного перемещения деталей соединения.

Разность размеров вала и отверстия до сборки, если размер вала больше размера отверстия, называется *натягом* N (рисунок 4.3, б):

$$N = d - D,$$

$$N_{\max} = d_{\max} - D_{\min} \text{ или } N_{\max} = es - EI,$$

$$N_{\min} = d_{\min} - D_{\max} \text{ или } N_{\min} = ei - ES. \quad (4.5)$$

Натяг характеризует степень сопротивления взаимному смещению деталей в соединении.

При соединении двух деталей образуется *посадка*, определяемая разностью их размеров до сборки, т. е. величиной получающихся зазоров или натягов в соединении. Посадка характеризует свободу относительного перемещения соединяемых деталей или степень сопротивления их взаимному перемещению.

В зависимости от взаимного расположения полей допусков отверстия или вала (по характеру соединения) посадка может быть (рисунок 4.4):

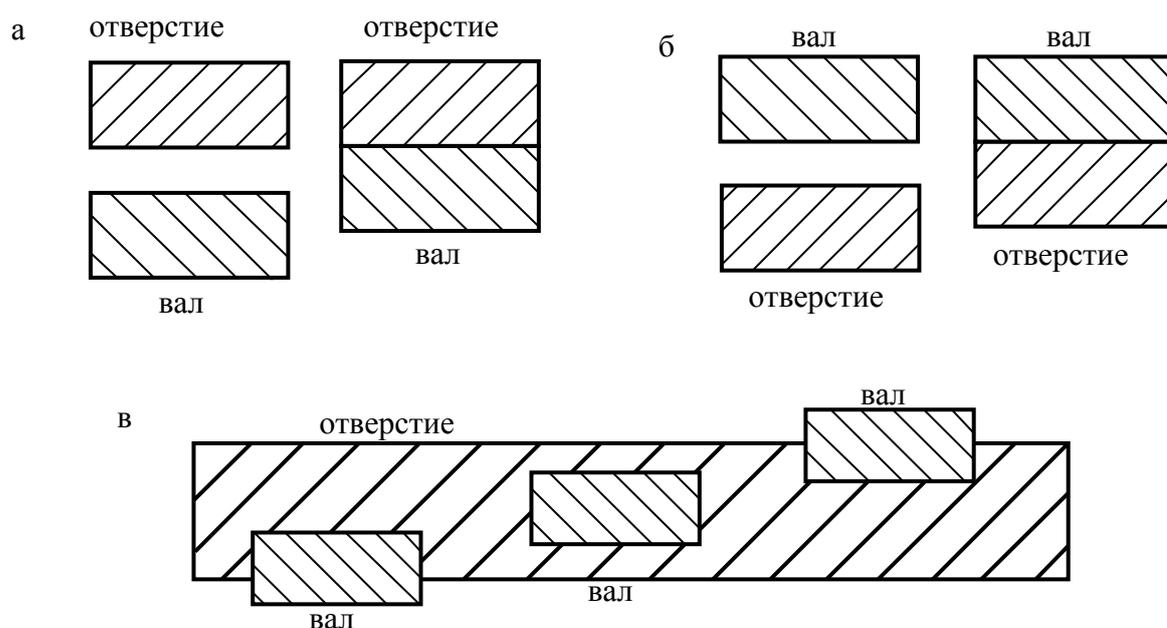


Рисунок 4.4 – Виды посадок

- подвижная (посадка с зазором): поле допуска отверстия над полем допуска вала (рисунок 4.4, а);
- неподвижная (посадка с натягом): поле допуска отверстия под полем допуска вала (рисунок 4.4, б);

– переходная (возможны натяг или зазор в зависимости от действительных размеров): поля допусков отверстия и вала перекрываются частично или полностью (рисунок 4.4, в).

Для сопрягаемых деталей установлены две системы расположения полей допусков:

- система отверстия;
- система вала.

Посадки в системе отверстия – посадки, в которых нужные зазоры и натяги получаются сочетанием различных полей допусков валов с одним и тем же полем допуска основного отверстия (рисунок 4.5, а).

Пример: $\varnothing 20 H7/f6$.

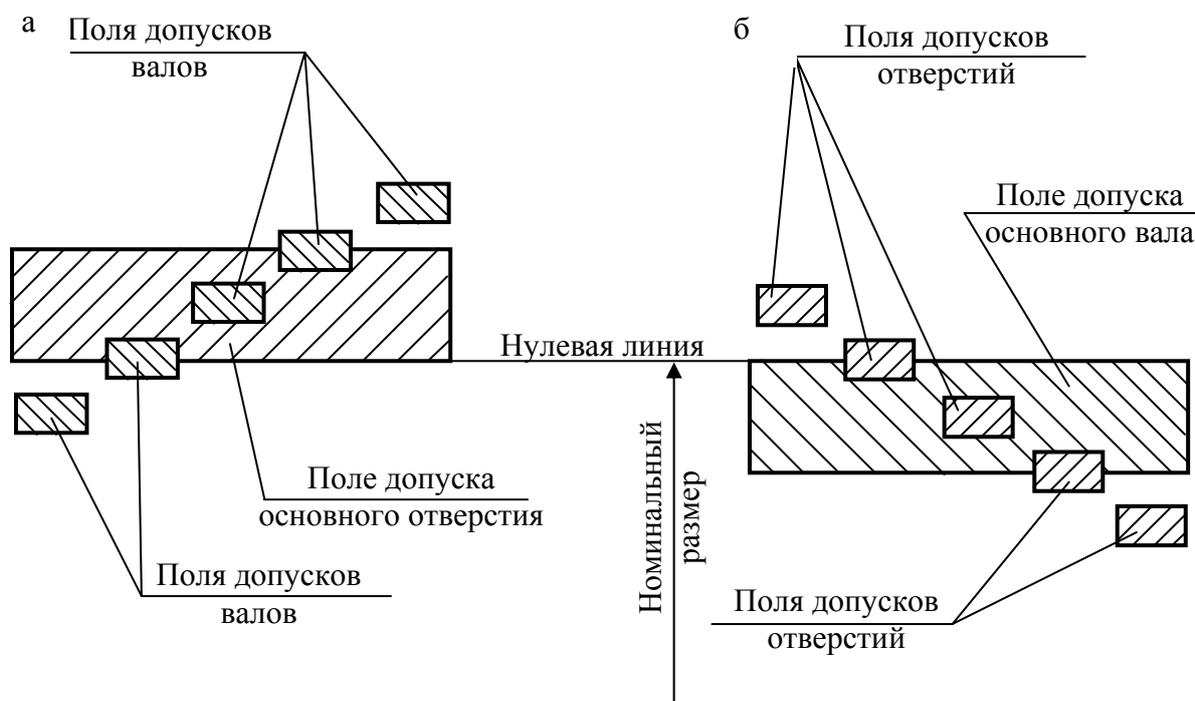


Рисунок 4.5 – Примеры посадок в системе отверстия и в системе вала

Применение системы отверстия предпочтительней.

Посадки в системе вала – посадки, в которых нужные зазоры и натяги получаются сочетанием различных полей допусков отверстий с одним и тем же полем допуска вала (рисунок 4.5, б).

Система вала применяется реже, например, если необходимо получить разные посадки нескольких деталей на одном гладком валу.

Пример: $\varnothing 20 K7/h6$.

Характеристикой расположения поля допуска в ЕСПД является числовое значение основного отклонения – того из двух предельных отклонений размера, которое находится ближе к нулевой линии. Для всех полей допусков, расположенных ниже нулевой линии, основным является верхнее отклонение es или ES (сокращение французского термина *ecart superieur* – верхнее отклонение); для полей допусков, расположенных выше нулевой линии, основным – нижнее отклонение ei или EI (сокращение французского термина *ecart inferieur* – нижнее отклонение) – (рисунок 4.6).

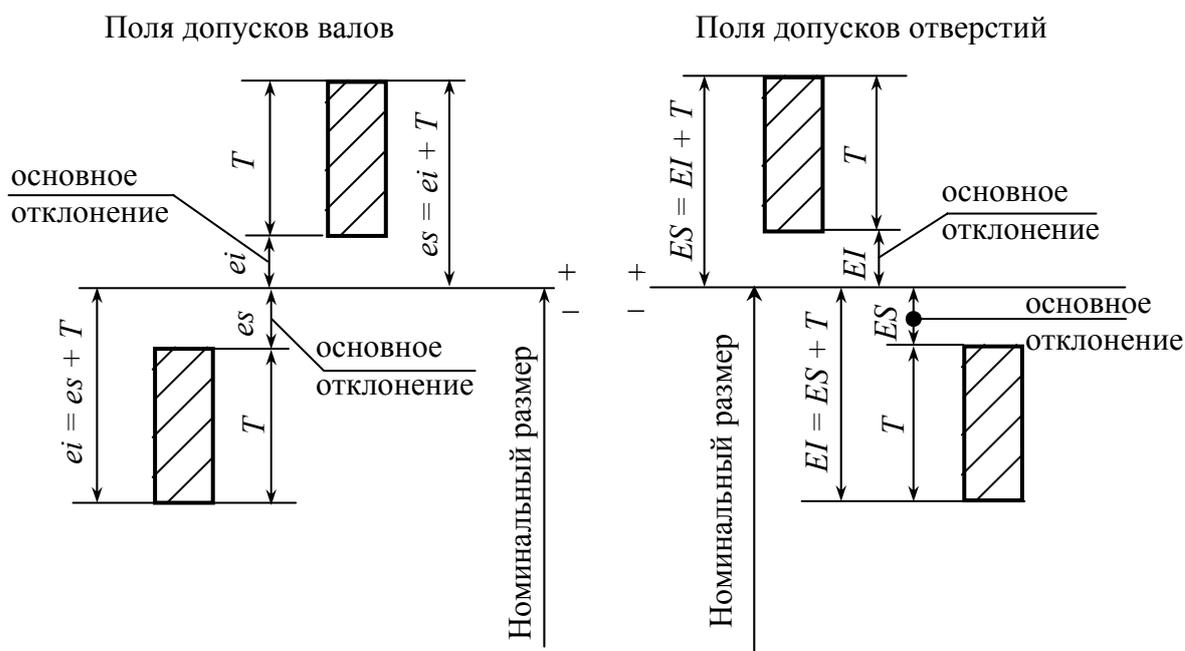


Рисунок 4.6 – Основные отклонения

Для удовлетворения требований в отношении отдельных деталей и их посадок для каждого номинального размера предусмотрены гаммы допусков и основных отклонений, характеризующих положение этих допусков относительно нулевой линии.

Допуск, величина которого зависит от номинального размера, обозначается цифрами (кавалитет).

Положение поля допуска относительно нулевой линии, зависящее от номинального размера, обозначается буквой латинского алфавита (или в некоторых случаях двумя буквами) – прописной для отверстий и строчной для валов (рисунок 4.7).

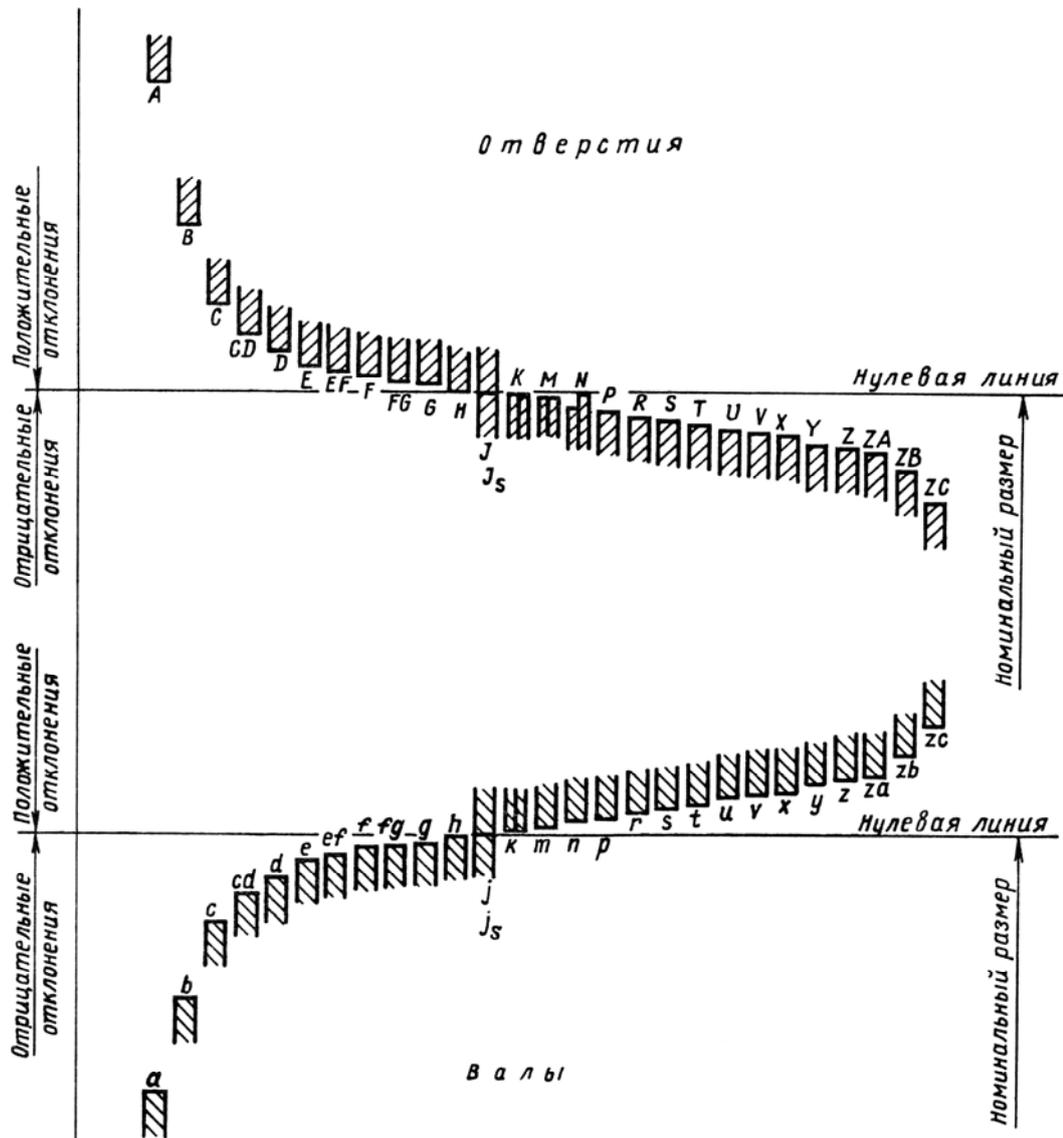


Рисунок 4.7 – Относительные положения полей допусков

Таким образом, размер, для которого указывается поле допуска, обозначается числом, за которым следует условное обозначение, состоящее из буквы (иногда из двух букв) и цифры (или двух цифр).

Примеры: 40 g6, 40 H7, 40 H11.

В обозначение посадки входит номинальный размер, общий для обоих соединяемых элементов (отверстия и вала), за которым следуют обозначения полей допусков для каждого элемента, начиная с отверстия. Условное обозначение посадки дается в виде дроби, причем в числителе указывают обозначение поля допуска отверстия, в знаменателе – обозначение поля допуска вала.

Пример: 40 H7/g6 (или $40 \frac{H7}{g6}$).

Таким образом, основные отклонения обозначают:

- для отверстий прописными: $A \dots ZC$;
- для валов строчными: $a \dots zc$.

Поле допуска – сочетанием букв основного отклонения и порядкового номера качества: g6, js7, H7, H11.

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАБОТЫ

5.1 Выполнение работы начинается с выбора посадки и номинального размера из приложения 1 по варианту, указанному преподавателем.

5.2 Задание 1

Используя таблицы ГОСТ 25347-82 (приложение 2), для заданных посадок определить:

- верхние и нижние предельные отклонения отверстия (ES, EI);
- верхние и нижние предельные отклонения вала (es, ei).

Применяя формулы 4.1–4.3, найти:

- наибольшие, наименьшие предельные размеры отверстия (D_{\max}, D_{\min});
- наибольшие, наименьшие предельные размеры вала (d_{\max}, d_{\min});

– допуски размеров деталей, входящих в соединение (отверстия T_D и вала T_d).

Пример: 24 H8/f7.

Отверстие $24_{0}^{+0,033}$ мм; вал $24_{-0,041}^{-0,020}$ мм.

$ES = +0,033$ мм, $EI = 0$ мм, $es = -0,020$ мм, $ei = -0,041$ мм.

$D_{\max} = 24,033$ мм, $D_{\min} = 24$ мм,

$d_{\max} = 23,980$ мм, $d_{\min} = 23,959$ мм.

$T_D = 0,033$ мм, $T_d = 0,021$ мм.

5.3 Задание 2

Построить схемы расположения полей допусков деталей, входящих в соединение. Определить, к какой системе относится заданная посадка.

Построение начинается с проведения нулевой линии, которая соответствует номинальному размеру деталей. Выше этой линии откладываются положительные отклонения, ниже – отрицательные. Далее полученные значения соединяются в произвольные прямоугольники (свое поле допуска для отверстия и для вала), заштриховываются и подписываются.

На примере той же посадки 24 H8/f7 построим схему расположения полей допусков (рисунок 5.1).

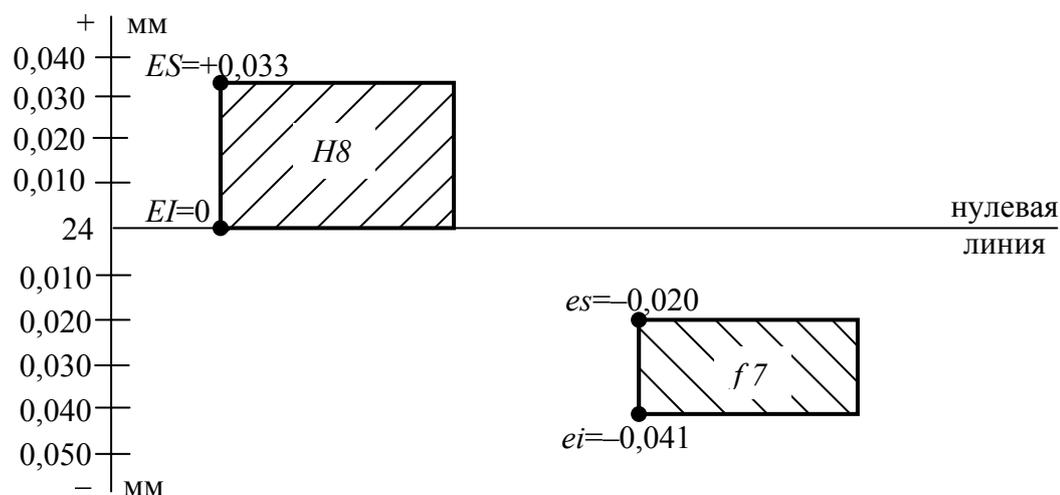


Рисунок 5.1 – Схема расположения полей допусков деталей

Согласно рисунку 4.5 посадка 24 H8/f7 относится к системе отверстия.

5.4 Задание 3

Определить тип посадки: с зазором, с натягом или переходная. Найти наибольшие и наименьшие зазоры и (или) натяги, показать их на рисунке.

Если поле допуска отверстия находится над полем допуска вала, то *посадка с зазором*. В этом случае определяются максимальный и минимальный зазоры S_{\max} , S_{\min} по формуле 4.4.

Если поле допуска отверстия находится под полем допуска вала – *посадка с натягом*. В этом случае определяются максимальный и минимальный натяги N_{\max} , N_{\min} по формуле 4.5.

Если поля допусков отверстия и вала перекрываются (пересекаются) частично или полностью (рис. 4.4 в), то посадка переходная (возможны и натяг и зазор). В этом случае определяются максимальные значения зазора и натяга S_{\max} , N_{\max} по тем же формулам.

Продолжим исследовать посадку $24\ H8/f7$. Глядя на рисунок 5.1, видим, что поле допуска отверстия $H8$ находится над полем допуска вала $f7$. Следовательно, посадка с зазором. Определим значения зазоров:

$$S_{\max} = D_{\max} - d_{\min} = 24,033 - 23,959 = 0,074 \text{ мм},$$

$$S_{\min} = D_{\min} - d_{\max} = 24,000 - 23,980 = 0,020 \text{ мм}.$$

Полученные значения показываем на рисунке 5.2

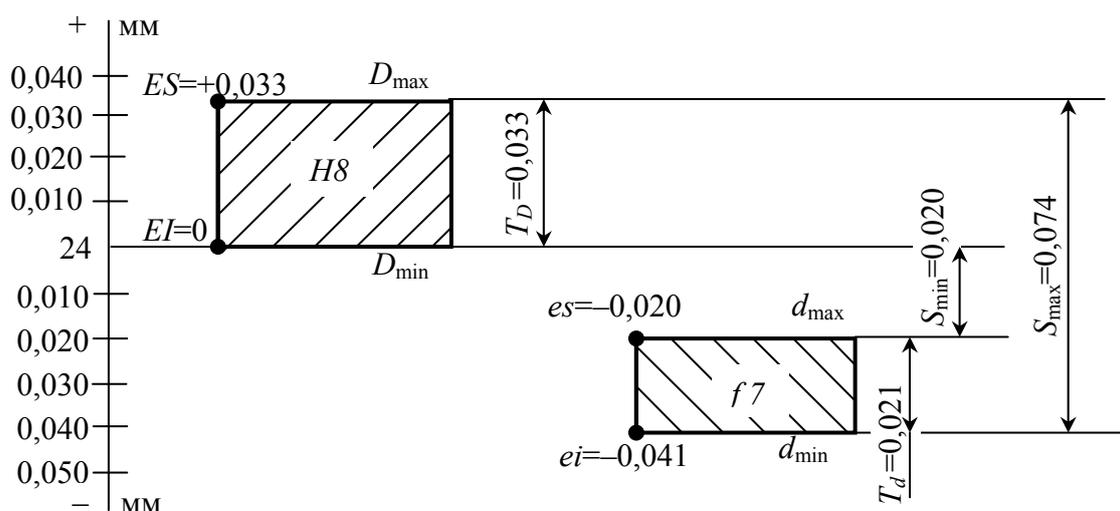


Рисунок 5.2 – Схема посадки $24\ H8/f7$

5.5 Задание 4

Полученные данные занести в таблицу 5.1.

Таблица 5.1

Таблица результатов, мм

Посадка	D_{\max}	D_{\min}	T_D	d_{\max}	d_{\min}	T_d	$T_D + T_d$	Зазоры		Натяги		Вид посадки
								S_{\max}	S_{\min}	N_{\max}	N_{\min}	

Заполняем таблицу согласно рассматриваемому примеру:

Посадка	D_{\max}	D_{\min}	T_D	d_{\max}	d_{\min}	T_d	$T_D + T_d$	Зазоры		Натяги		Вид посадки
								S_{\max}	S_{\min}	N_{\max}	N_{\min}	
24H8/f7	24,033	24,000	0,033	23,980	23,959	0,021	0,054	0,074	0,020	–	–	Посадка с зазором

6 СОДЕРЖАНИЕ ОТЧЕТА

6.1 Отчет оформляется на 4-х страницах формата А4.

6.2 На стр. 1 помещается титульный лист согласно установленному образцу.

6.3 На стр. 2 располагаются следующие данные:

- цель работы;
- задание на лабораторную работу;
- необходимые приборы и материалы;
- исходные данные (выбранные по варианту посадки);

6.4 На стр. 3 располагаются схемы полей допусков деталей с указанием:

- номинального размера;
- верхних и нижних предельных отклонений;
- допусков на размер;
- максимальных и минимальных зазоров или натягов в зависимости от типа посадки.

6.5 На стр. 4 должны быть:

- табл. 1 с результатами вычисленных величин;
- выводы.

7 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое допуск размера и допуск посадки?
2. Чем характеризуется поле допуска?
3. В каких единицах изображаются допустимые отклонения на чертежах, а в каких – в таблицах полей допусков?
4. Что называется системой отверстия и системой вала? Какая система предпочтительней?
5. Как обозначается посадка?
6. Сколько квалитетов установлено стандартом?
7. Как различаются посадки по характеру соединения (3 вида)?
8. Что называется номинальным, действительным и предельным размерами?
9. Что называется нижним и верхним предельными отклонениями?
10. Что такое зазор и каково его назначение в сопряжении?
11. Что называют натягом и каково его назначение в сопряжениях деталей?
12. Чем характеризуется точность размера?
13. Есть ли связь между точностью размеров и шероховатостью поверхности?

СПИСОК ЛИТЕРАТУРЫ

1. Белкин И. М. Допуски и посадки. – М. : Машиностроение, 1992. – 340 с.
2. Ганевский Г. М., Гольдин И. И. Допуски, посадки и технические измерения в машиностроении. – М. : Высш. шк., 1998. – 288 с.
3. Допуски и посадки: Справочник / под ред. В. Д. Мягкова. Ч.1. – Л. : Машиностроение, 1982. – 544 с.
4. Допуски и посадки: Справочник / под ред. В. Д. Мягкова. Ч. 2 – Л. : Машиностроение, 1982. – 448 с.
5. Якушев А. И., Воронцов Л. Н., Федотов Н. М. Взаимозаменяемость, стандартизация и технические измерения. – М. : Машиностроение, 1986. – 352 с.

Приложение А

Таблица 1

Исходные данные

№ варианта	Посадки			
1	250 H7/e8	40 H6/n6	62 H7/s6	32 F7/h5
2	315 H7/c8	24 H8/k7	50 H7/s7	48 N7/h6
3	400 H8/d8	32 H6/k5	60 H8/u8	80 G6/h5
4	200 H7/e7	50 H8/j _s 7	42 H6/s5	10 K6/h5
5	105 H7/f7	8 H7/m6	40 H8/x8	24 K7/h6
6	30 H6/f7	48 H7/n7	12 H7/s6	120 E8/h7
7	120 E8/h7	12 H6/j _s 5	42 H7/t6	80 P7/h7
8	60 H6/g5	8 H7/n6	32 H8/s7	120 H7/h6
9	140 H7/g6	24 H6/j _s 5	42 H8/x8	12 M6/h5
10	10 H5/g4	24 H6/k5	80 H8/s7	8 J _s 7/h6
11	42 H6/g5	5 H5/k4	50 H8/z8	80 F8/h6
12	220 H8/d9	12 H7/n6	40 H6/s5	105 D8/h6
13	400 H8/e8	10 H5/n4	52 H6/s5	120 E8/h6
14	120 H8/f8	48 H6/n5	62 H7/s5	8 P7/h6
15	20 H7/g6	45 H6/m5	60 H6/p5	80 D9/h9
16	120 H8/f7	12 H6/k5	32 H7/u7	60 F8/h6
17	315 H9/d9	48 H7/k6	32 H6/s5	10 J _s 8/h7
18	140 H8/d8	8 H5/m4	62 H7/p5	48 F8/h7
19	105 H9/n6	6 H5/m4	36 H7/u7	54 H8/h8
20	80 H9/d9	16 H7/k6	48 H8/s7	12 H5/h4
21	400 H11/d11	32 H6/j _s 5	60 H7/s6	120 F8/h6
22	80 H8/h7	12 H5/m4	48 H7/p6	62 F7/h6
23	82 H8/f9	40 H7/k6	120 H6/p5	12 K7/h6
24	140 H9/d9	8 H6/j _s 5	80 H8/x8	36 S7/h6
25	64 H8/h8	20 H7/n6	52 H8/s7	8 M6/h5
26	36 H8/h8	12 H7/j _s 6	50 H7/p6	52 P6/h5
27	400 H11/d11	120 H6/p5	32 H7/u7	10 H5/n4
28	10 H5/n4	120 E8/h7	80 H9/d9	52 H6/s5
29	8 H7/n6	48 H7/n7	62 F7/h6	105 D8/h6
30	5 H5/k4	250 H7/e8	10 J _s 8/h7	140 H7/g6
31	315 H9/d9	50 H8/z8	105 D8/h6	6 H5/m4
32	32 H7/u7	6 H5/m4	120 E8/h7	80 H9/d9
33	52 H6/s5	400 H11/d11	5 H5/k4	250 H7/e8
34	140 H7/g6	10 J _s 8/h7	50 H8/z8	48 H7/n7
35	62 F7/h6	8 H7/n6	315 H9/d9	120 H6/p5
36	12 H5/m4	82 H8/f9	40 H8/x8	400 H8/e8
37	140 H8/d8	40 H7/k6	315 H9/d9	32 F7/h5
38	40 H8/x8	250 H7/e8	50 H8/z8	12 H5/m4
39	12 H6/k5	8 H5/m4	120 H8/f7	48 N7/h6
40	400 H9/e8	16 H5/j _s 6	62 H7/s6	30 H7/h5
41	450 H11/d11	52 H7/k6	32 H8/u8	42 H8/h8

Продолжение табл. 1

№ варианта	Посадки			
42	250 H7/e8	24 H8/k7	22 H7/s6	10 K6/h5
43	105 H7/f7	48 H7/n6	160 H7/t6	12 M6/h5
44	42 H7/h6	10 H5/j _s 4	32 H6/s6	16 K8/h7
45	80 H8/e9	32 H6/j _s 5	48 H6/p5	24 K7/h6
46	32 H8/h9	8 H5/n4	60 H7/t6	16 F8/h6
47	124 H12/h12	16 H6/m5	24 H8/x8	32 N6/h5
48	42 H11/h11	24 H7/k6	30 H7/p6	40 N8/h7
49	64 H10/d10	18 H5/m4	124 H7/s6	32 K7/h6
50	24 H9/h8	40 H6/k5	82 H7/s6	140 E8/h8
51	72 H9/e8	12 H5/k4	42 H7/s7	64 T7/h6
52	220 H12/b12	22 H5/k4	64 H7/s7	16 M8/h7
53	80 H10/h9	42 H5/k4	24 H7/p6	124 F7/h6
54	105 H9/e9	22 H6/k5	84 H7/p6	8 M5/h4
55	84 H11/c11	6 H6/k5	24 H8/u8	42 P6/h5
56	32 H9/h9	24 H8/n7	64 H6/s5	105 E9/h9
57	124 H10/h10	12 H8/n7	48 H6/s5	60 F8/h7
58	160 H11/d11	8 H8/m7	36 H8/z8	24 J _s 6/h5
59	48 H9/e9	18 H8/m7	105 H8/z8	60 N7/h6
60	64 H8/e8	24 H7/m6	96 H6/s5	110 H11/h11
61	105 H7/e7	60 H8/m7	16 H6/s5	48 H8/h7
62	24 H5/h4	64 H6/j _s 5	110 H7/s7	140 D9/h8
63	48 H5/h4	24 H6/n5	120 H8/u8	164 E8/h6
64	10 H5/g4	60 H7/t6	140 H7/g6	24 K7/h6
65	42 H6/g5	24 H8/x8	6 H5/m4	16 F8/h6
66	220 H8/d9	30 H7/p6	80 H9/d9	32 N6/h5
67	400 H8/e8	124 H7/s6	250 H7/e8	40 N8/h7
68	120 H8/f8	82 H7/s6	48 H7/n7	32 K7/h6
69	20 H7/g6	42 H7/s7	120 H6/p5	140 E8/h8
70	20 H7/n6	80 H9/d9	160 H11/d11	32 F7/h5
71	12 H7/j _s 6	62 F7/h6	48 H9/e9	48 N7/h6
72	120 H6/p5	10 J _s 8/h7	64 H8/e8	80 G6/h5
73	120 E8/h7	105 D8/h6	105 H7/e7	10 K6/h5
74	48 H7/n7	120 E8/h7	24 H5/h4	24 K7/h6
75	102 E8/h7	15 D8/h6	305 H7/e7	70 K6/h5
76	23 H9/h9	94 H8/n7	364 H6/s5	105 E9/h9
77	64 H8/e8	44 H7/m6	6 H6/s5	310 H11/h11
78	80 H10/h9	41 H5/k4	204 H7/p6	12 F7/h6
79	204 H5/h4	14 H6/j _s 5	90 H7/s7	140 D9/h8
80	200 H7/g6	4 H7/s7	121 H6/p5	40 E8/h8
81	48 H6/g5	104 H8/x8	206 H5/m4	16 F8/h6
82	280 H12/b12	2 H5/k4	64 H7/s7	116 M8/h7
83	400 H5/h4	84 H6/n5	12 H8/u8	164 E8/h6
84	16 H11/d11	308 H8/m7	136 H8/z8	4 J _s 6/h5

Продолжение табл. 1

№ варианта	Посадки			
	85	10 H5/g4	66 H7/t6	340 H7/g6
86	408 H7/n7	125 E8/h7	4 H5/h4	84 K7/h6
87	28 H7/n6	480 H9/d9	180 H11/d11	3 F7/h5
88	122 H8/d9	38 H7/p6	280 H9/d9	3 N6/h5
89	201 H8/f8	88 H7/s6	148 H7/n7	32 K7/h6
90	40 H8/e8	14 H7/s6	250 H7/e8	40 N8/h7
91	84 H11/c11	36 H6/k5	324 H8/u8	4 P6/h5
92	21 H7/j _s 6	262 F7/h6	348 H9/e9	8 N7/h6
93	10 H7/e7	300 H8/m7	106 H6/s5	68 H8/h7
94	214 H10/h10	120 H8/n7	8 H6/s5	60 F8/h7
95	48 H9/e9	180 H8/m7	15 H8/z8	360 N7/h6
96	72 H9/e8	112 H5/k4	402 H7/s7	6 T7/h6
97	15H9/e9	202 H6/k5	84 H7/p6	8 M5/h4
98	12 H6/p5	70 J _s 8/h7	364 H8/e8	80 G6/h5
99	330 H6/f7	56 H6/m5	12 H7/s6	120 E8/h7
100	124 H12/h12	48 H7/n7	24 H8/x8	3 N6/h5

Приложение Б

ЕДИНАЯ СИСТЕМА ДОПУСКОВ И ПОСАДОК

**ПОЛЯ ДОПУСКОВ
И РЕКОМЕНДУЕМЫЕ ПОСАДКИ**

**ГОСТ 25347-82
(СТ СЭВ 144-75)**

Квалитеты 4 и 5

Интервал размеров, мм	Поля допусков														
	g4	h4	j _s 4	k4	m4	n4	g5	h5	j _s 5	k5	m5	n5	p5	r5	s5
	Предельные отклонения, мкм														
От 1 до 3	-2	0	+1,5	+3	+5	+7	-2	0	+2,0	+4	+6	+8	+10	+14	+18
	-5	-3	-1,5	0	+2	+4	-6	-4	-2,0	0	+2	+4	+6	+10	+14
Свыше 3 до 6	-4	0	+2,0	+5	+8	+12	-4	0	+2,5	+6	+9	+13	+17	+20	+24
	-8	-4	-2,0	+1	+4	+8	-9	-5	-2,5	+1	+4	+8	+12	+15	+19
Свыше 6 до 10	-5	0	+2,0	+5	+10	+14	-5	0	+3,0	+7	+12	+16	+21	+25	+29
	-9	-4	-2,0	+1	+6	+10	-11	-6	-3,0	+1	+6	+10	+15	+19	+23
Свыше 10 до 14	-6	0	+2,5	+6	+12	+17	-6	0	+4,0	+9	+15	+20	+26	+31	+36
	-11	-5	-2,5	+1	+7	+11	-14	-8	-4,0	+1	+7	+12	+18	+23	+28
Свыше 14 до 18	-7	0	+3,0	+8	+14	+21	-7	0	+4,5	+11	+17	+24	+31	+37	+44
	-13	-6	-3,0	+2	+8	+15	-16	-9	-4,5	+2	+8	+15	+22	+18	+35
Свыше 18 до 24	-9	0	+3,5	+9	+16	+24	-9	0	+5,5	+13	+20	+28	+37	+45	+54
	-16	-7	-3,5	+2	+9	+17	-20	-11	-5,5	+2	+9	+17	+26	+34	+43
Свыше 24 до 30	-10	0	+4,0	+10	+19	+28	-10	0	+6,5	+15	+24	+33	+45	+54	+66
	-18	-8	-4,0	+2	+11	+20	-23	-13	-6,5	+2	+11	+20	+32	+41	+53
Свыше 30 до 40	-12	0	+5,0	+13	+23	+33	-12	0	+7,5	+18	+28	+38	+52	+66	+86
	-22	-10	-5,0	+3	+13	+23	-27	-15	-7,5	+3	+13	+23	+37	+51	+71
Свыше 40 до 50	-14	0	+6,0	+15	+27	+39	-14	0	+9,0	+21	+33	+45	+61	+81	+110
	-26	-12	-6,0	+3	+15	+27	-32	-18	-9,0	+3	+15	+27	+43	+63	+92
Свыше 50 до 65	-15	0	+7,0	+18	+31	+45	-15	0	+10,0	+24	+37	+51	+70	+97	+142
	-29	-14	-7,0	+4	+17	+31	-35	-20	-10,0	+4	+17	+31	+50	+77	+122
Свыше 65 до 80	-17	0	+8,0	+20	+36	+50	-17	0	+11,5	+27	+43	+57	+79	+117	+181
	-33	-16	-8,0	+4	+20	+34	-40	-23	-11,5	+4	+20	+34	+56	+94	+158
Свыше 80 до 100	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+108	+190
Свыше 100 до 120	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+126	+232
Свыше 120 до 140	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+139	+233
Свыше 140 до 160	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+159	+279
Свыше 160 до 180	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+139	+233
Свыше 180 до 200	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+159	+279
Свыше 200 до 225	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+139	+233
Свыше 225 до 250	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+159	+279
Свыше 250 до 280	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+139	+233
Свыше 280 до 315	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+159	+279
Свыше 315 до 355	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+139	+233
Свыше 355 до 400	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+159	+279
Свыше 400 до 450	-18	0	+9,0	+22	+39	+55	-18	0	+12,5	+29	+46	+62	+87	+133	+215
	-36	-18	-9,0	+4	+11	+37	-43	-25	-12,5	+4	+21	+37	+62	+139	+233
Свыше 450 до 500	-20	0	+10,0	+25	+43	+60	-20	0	+13,5	+32	+50	+67	+95	+153	+259
	-40	-20	-10,0	+5	+23	+40	-47	-27	-13,5	+5	+23	+40	+68	+159	+279

Квалитет 6

Интервал размеров, мм	Поля допусков										
	<i>f</i> 6	<i>g</i> 6	<i>h</i> 6	<i>j</i> _s 6	<i>k</i> 6	<i>m</i> 6	<i>n</i> 6	<i>p</i> 6	<i>r</i> 6	<i>s</i> 6	<i>t</i> 6
	Предельные отклонения, мкм										
От 1 до 3	-6	-2	0	+3,0	+6	+8	+10	+12	+16	+20	-
	-12	-8	-6	-3,0	0	+1	+4	+6	+10	+14	-
Свыше 3 до 6	-10	-4	0	+4,0	+9	+12	+16	+10	+23	+27	-
	-18	-12	-8	-4,0	+1	+4	+8	+12	+15	+19	-
Свыше 6 до 10	-13	-5	0	+4,5	+10	+15	+19	+24	+28	+32	-
	-22	-14	-9	-4,5	+1	+6	+10	+15	+19	+23	-
Свыше 10 до 14	-16	-6	0	+5,5	+12	+18	+23	+29	+34	+39	-
Свыше 14 до 18	-27	-17	-11	-5,5	+1	+7	+12	+18	+23	+28	-
Свыше 18 до 24	-20	-7	0	+6,5	+15	+21	+28	+35	+41	+48	-
Свыше 24 до 30	-33	-20	-13	-6,5	+2	+8	+15	+22	+28	+35	+54 +41
Свыше 30 до 40	-25	-9	0	+8,0	+18	+25	+33	+42	+50	+59	+64 +48
Свыше 40 до 50	-41	-25	-16	-8,0	+1	+9	+17	+26	+34	+43	+70 +54
Свыше 50 до 65	-30	-10	0	+9,5	+21	+30	+39	+51	+60	+72	+85
Свыше 65 до 80	-49	-29	-19	-9,5	+1	+11	+20	+32	+41	+53	+66
Свыше 80 до 100	-36	-12	0	+11,0	+25	+35	+45	+59	+62	+78	+94
Свыше 100 до 120	-56	-34	-22	-11,0	+3	+13	+23	+37	+43	+59	+75
Свыше 120 до 140									+73	+93	+113
Свыше 140 до 160	-43	-14	0	+12,5	+28	+40	+52	+68	+51	+71	+91
Свыше 160 до 180	-68	-39	-25	-12,5	+3	+15	+27	+43	+76	+101	+126
Свыше 180 до 200									+54	+79	+104
Свыше 200 до 225	-50	-15	0	+14,5	+33	+46	+60	+79	+88	+117	+147
Свыше 225 до 250	-79	-44	-29	-14,5	+4	+17	+31	+50	+63	+92	+122
Свыше 250 до 280	-56	-17	0	+16,0	+36	+52	+66	+88	+90	+125	+159
Свыше 280 до 315	-88	-49	-32	-16,0	+4	+20	+34	+56	+65	+100	+154
Свыше 315 до 355	-62	-18	0	+18,0	+40	+57	+73	+98	+93	+133	+171
Свыше 355 до 400	-98	-54	-36	-18,0	+4	+21	+57	+62	+68	+108	+146
Свыше 400 до 450	-68	-20	0	+20,0	+45	+63	+80	+108	+106	+151	+195
Свыше 450 до 500	-108	-60	-40	-20,0	+5	+23	+40	+68	+77	+122	+166
									+109	+159	+209
									+80	+130	+180
									+113	+169	+226
									+84	+140	+196
									+126	+190	+250
									+94	+158	+218
									+130	+202	+272
									+98	+170	+240
									+144	+226	+304
									+108	+190	+268
									+150	+244	+330
									+114	+208	+294
									+166	+272	+370
									+126	+232	+330
									+172	+292	+400
									+132	+252	+360

Квалитет 7

Интервал размеров, мм	Поля допусков								
	<i>e7</i>	<i>f7</i>	<i>h7</i>	<i>js7</i>	<i>k7</i>	<i>m7</i>	<i>n7</i>	<i>s7</i>	<i>u7</i>
	Предельные отклонения, мкм								
От 1 до 3	-14	-6	0	+5	+10	-	+14	+24	+28
	-24	-16	-10	-5	0	-	+4	+14	+18
Свыше 3 до 6	-20	-12	0	+6	+13	+16	+20	+31	+35
	-32	-22	-12	-6	+1	+4	+8	+19	+23
Свыше 6 до 10	-25	-13	0	+7	+16	+21	+25	+38	+43
	-40	-28	-15	-7	+1	+6	+10	23	+28
Свыше 10 до 14	-32	-16	0	+9	+19	+25	+30	+45	+51
Свыше 14 до 18	-50	-34	-18	-9	+1	+7	+12	+28	+33
Свыше 18 до 24	-40	-20	0	+10	+23	+29	+36	+56	+62
Свыше 24 до 30	-61	-41	-21	-10	+2	+8	+15	+35	+41
Свыше 30 до 40	-50	-25	0	+12	+27	+34	+42	+68	+69
Свыше 40 до 50	-75	-50	-25	-12	+2	+9	+17	+43	+48
Свыше 50 до 65	-60	-30	0	+15	+32	+41	+50	+83	+85
Свыше 65 до 80	-90	-60	-30	-15	+2	+11	+20	+53	+60
Свыше 80 до 100	-72	-36	0	+17	+38	+48	+58	+71	+87
Свыше 100 до 120	-107	-71	-35	-17	+3	+13	+23	+89	+117
Свыше 120 до 140								+59	+132
Свыше 140 до 160	-85	-43	0	+20	+43	+55	+67	+106	+117
Свыше 160 до 180	-125	-83	-40	-20	+3	+15	+27	+71	+87
Свыше 180 до 200								+132	+132
Свыше 200 до 225	-100	-50	0	+23	+50	+63	+77	+140	+132
Свыше 225 до 250	-145	-96	-46	-23	+4	+17	+31	+92	+170
Свыше 250 до 280	-110	-56	0	+26	+56	+72	+86	+148	+210
Свыше 280 до 315	-162	-108	-52	-26	+4	+20	+34	+108	+210
Свыше 315 до 355	-125	-62	0	+28	+61	+78	+94	+168	+282
Свыше 355 до 400	-182	-119	-57	-28	+4	+21	+37	+122	+236
Свыше 400 до 450	-135	-68	0	+31	+68	+86	+103	+176	+304
Свыше 450 до 500	-198	-131	-63	-31	+5	+23	+40	+130	+259
								+186	+330
								+140	+284
								+210	+367
								+158	+315
								+222	+402
								+170	+350
								+247	+447
								+190	+390
								+265	+492
								+208	+435
								+295	+553
								+232	+490
								+315	+603
								+252	+540

Квалитеты 8 и 9

Интервал размеров, мм	Поля допусков													
	<i>c8</i>	<i>d8</i>	<i>e8</i>	<i>f8</i>	<i>h8</i>	<i>js8*</i>	<i>u8</i>	<i>x8</i>	<i>z8</i>	<i>d9</i>	<i>e9</i>	<i>f9</i>	<i>h9</i>	<i>js9*</i>
	Предельные отклонения, мкм													
От 1 до 3	-60	-20	-14	-6	0	+7	+32	+34	+40	-20	-14	-6	0	+12
	-74	-34	-28	-20	-14	-7	+18	+20	+25	-45	-39	-31	-25	-12
Свыше 3 до 6	-70	-30	-20	-10	0	+9	+41	+46	+53	-30	-20	-10	0	+15
	-88	-48	-38	-28	-18	-9	+23	+28	-35	-60	-50	-40	-30	-15
Свыше 6 до 10	-80	-40	-25	-13	0	+11	+50	+56	+64	-40	-25	-13	0	+18
	-102	-62	-47	-35	-22	-11	+28	+34	+42	-76	-61	-49	-36	-18
Свыше 10 до 14	-95	-50	-32	-16	0	+13	+60	+67	+77	-50	-32	-16	0	+21
	-122	-77	-59	-43	-27	-13	+33	+40	+50	-93	-75	-59	-43	-21
Свыше 14 до 18								+72	+87					
								+45	+60					
Свыше 18 до 24	-110	-65	-40	-20	0	+16	+74	+87	+106	-65	-40	-20	0	+26
	-143	-93	-73	-53	-33	-16	+41	+54	+73	-117	-92	-72	-52	-26
Свыше 24 до 30							+81	+97	+121					
							+48	+64	+88					
Свыше 30 до 40	-120				0	+19	+99	+119	+151	-80	-50	-25	0	+31
	-159	-80	-50	-25	0	+19	+60	+80	+112	-142	-112	-87	-62	-31
Свыше 40 до 50	-130	-119	-89	-64	-39	-19	+109	+136	+175					
	-169						+70	+97	+136					
Свыше 50 до 65	-140				0	+23	+133	+168	+218	+100	-60	-30	0	+37
	-186	-100	-60	-30	0	+23	+87	+122	+172	-174	-134	-104	-74	-37
Свыше 65 до 80	-150	-146	-106	-76	-46	-23	+148	+192	+256					
	-196						+102	+146	+210					
Свыше 80 до 100	-170				0	+27	+178	+232	+312	-120	-72	-36	0	+43
	-224	-120	-72	-36	0	+27	+124	+178	+258	-207	-159	-123	-87	-43
Свыше 100 до 120	-180	-174	-126	-90	-54	-27	+198	+264	+354					
	-234						+144	+210	+310					
Свыше 120 до 140	-200						+233	+311	+428					
	-283						+170	+248	+355					
Свыше 140 до 160	-210	-145	-85	-43	0	+31	+253	+343	+478	-145	-85	-43	0	+50
	-273	-208	-148	-105	-63	-31	+190	+280	+415	-245	-185	-143	-100	-50
Свыше 160 до 180	-230						+273	+373	+528					
	-293						+210	+310	+465					
Свыше 180 до 200	-240						+308	+422	+592					
	-312						+236	+350	+520					
Свыше 200 до 225	-260	-170	-100	-50	0	+36	+330	+457	+647	-170	-100	-50	0	+57
	-332	-242	-172	-122	-72	-36	+258	+385	+575	-285	-215	-165	-115	-57
Свыше 225 до 250	-280						+356	+497	+712					
	-352						+284	+425	+640					
Свыше 250 до 280	-300				0	+40	+396	+556	+791	-190	-110	-56	0	+65
	-381	-190	-110	-56	0	+40	+315	+475	+710	-320	-240	-186	-130	-65
Свыше 280 до 315	-330	-271	-191	-137	-81	-40	+431	+606	+871					
	-411						+350	+525	+790					
Свыше 315 до 355	-360				0	+44	+479	+679	+989	-210	-125	-62	0	+70
	-499	-210	-125	-62	0	+44	+390	+590	+900	-350	-265	-202	-140	-70
Свыше 355 до 400	-400	-299	-214	-151	-89	-44	+524	+749	+1089					
	-489						+435	+660	+1000					
Свыше 400 до 450	-440				0	+48	+587	+837	+1197	-230	-135	-68	0	+77
	-537	-230	-135	-68	0	+48	+490	+740	+1100	-385	-290	-223	-155	-77
Свыше 450 до 500	-480	-327	-232	-165	-97	-48	+637	+917	+1347					
	-577						+540	+820	+1250					

Квалитеты от 10 до 12

Интервал размеров, мм	Поля допусков											
	<i>d</i> 10	<i>h</i> 10	<i>j_s</i> 10*	<i>a</i> 11	<i>b</i> 11	<i>c</i> 11	<i>d</i> 11	<i>h</i> 11	<i>j_s</i> 11*	<i>b</i> 12	<i>h</i> 12	<i>j_s</i> 12*
	Предельные отклонения, мкм											
От 1 до 3	-20	0	+20	-270	-140	-60	-20	0	+30	-140	0	+50
	-60	-40	-20	-330	-200	-120	-80	-60	-30	-240	-100	-50
Свыше 3 до 6	-30	0	+24	-270	-140	-70	-30	0	+37	-140	0	+60
	-78	-48	-24	-345	-215	-145	-105	-75	-37	-260	-120	-60
Свыше 6 до 10	-40	0	+29	-280	-150	-80	-40	0	+45	-150	0	+75
	-98	-58	-29	-370	-240	-170	-130	-90	-45	-300	-150	-75
Свыше 10 до 14	-50	0	+35	-290	-150	-95	-50	0	+55	-150	0	+90
	-120	-70	-35	-400	-260	-205	-160	-110	-55	-330	-180	-90
Свыше 14 до 18	-65	0	+42	-300	-160	-110	-65	0	+65	-160	0	+105
	-149	-84	-42	-430	-290	-240	-195	-130	-65	-370	-210	-105
Свыше 18 до 24	-80	0	+50	-310	-170	-120	-80	0	+80	-170	0	+125
	-180	-100	-50	-470	-330	-280	-240	-160	-80	-420	-250	-125
Свыше 24 до 30	-320	-180	-130	-320	-180	-130	-240	-160	-80	-180	-250	-125
	-480	-340	-290	-480	-340	-290	-480	-340	-290	-430	-290	-150
Свыше 30 до 40	-100	0	+60	-340	-190	-140	-100	0	+95	-190	0	+150
	-220	-120	-60	-530	-380	-330	-290	-190	-95	-490	-300	-150
Свыше 40 до 50	-360	-200	-150	-360	-200	-150	-290	-190	-95	-200	-300	-150
	-550	-390	-340	-550	-390	-340	-550	-390	-340	-500	-500	-500
Свыше 50 до 65	-120	0	+70	-380	-220	-170	-120	0	+110	-220	0	+175
	-260	-140	-70	-600	-440	-390	-340	-220	-110	-570	-350	-175
Свыше 65 до 80	-410	-240	-180	-410	-240	-180	-340	-220	-110	-240	-350	-175
	-630	-460	-400	-630	-460	-400	-630	-460	-400	-590	-590	-590
Свыше 80 до 100	-460	-260	-200	-460	-260	-200	-145	0	+125	-260	0	+200
	-710	-510	-450	-710	-510	-450	-395	-250	-125	-660	-400	-200
Свыше 100 до 120	-520	-280	-210	-520	-280	-210	-145	0	+125	-280	0	+200
	-770	-530	-460	-770	-530	-460	-395	-250	-125	-680	-400	-200
Свыше 120 до 140	-580	-310	-230	-580	-310	-230	-145	0	+125	-310	0	+200
	-830	-560	-480	-830	-560	-480	-395	-250	-125	-710	-400	-200
Свыше 140 до 160	-660	-340	-240	-660	-340	-240	-170	0	+145	-340	0	+230
	-950	-630	-530	-950	-630	-530	-460	-290	-145	-800	-460	-230
Свыше 160 до 180	-740	-380	-260	-740	-380	-260	-170	0	+145	-380	0	+230
	-1030	-670	-550	-1030	-670	-550	-460	-290	-145	-840	-460	-230
Свыше 180 до 200	-820	-420	-280	-820	-420	-280	-170	0	+145	-420	0	+230
	-1110	-710	-570	-1110	-710	-570	-460	-290	-145	-880	-460	-230
Свыше 200 до 225	-920	-480	-300	-920	-480	-300	-190	0	+160	-480	0	+260
	-1240	-800	-620	-1240	-800	-620	-510	-320	-160	-1000	-520	-260
Свыше 225 до 250	-1050	-540	-330	-1050	-540	-330	-190	0	+160	-540	0	+260
	-1370	-860	-650	-1370	-860	-650	-510	-320	-160	-1060	-520	-260
Свыше 250 до 280	-1200	-600	-360	-1200	-600	-360	-210	0	+180	-600	0	+285
	-1560	-960	-720	-1560	-960	-720	-570	-360	-180	-1170	-570	-285
Свыше 280 до 315	-1350	-680	-400	-1350	-680	-400	-210	0	+180	-680	0	+285
	-1710	-1040	-760	-1710	-1040	-760	-570	-360	-180	-1250	-570	-285
Свыше 315 до 355	-1500	-760	-440	-1500	-760	-440	-230	0	+200	-760	0	+315
	-1900	-1160	-840	-1900	-1160	-840	-630	-400	-200	-1390	-630	-315
Свыше 355 до 400	-1650	-840	-480	-1650	-840	-480	-230	0	+200	-840	0	+315
	-2050	-1240	-880	-2050	-1240	-880	-630	-400	-200	-1470	-630	-315
Свыше 400 до 450	-230	0	+125	-1500	-760	-440	-230	0	+200	-760	0	+315
	-480	-250	-125	-1900	-1160	-840	-630	-400	-200	-1390	-630	-315
Свыше 450 до 500	-1650	-840	-480	-1650	-840	-480	-230	0	+200	-840	0	+315
	-2050	-1240	-880	-2050	-1240	-880	-630	-400	-200	-1470	-630	-315

Квалитеты от 13 до 17

Интервал размеров, мм	Поля допусков									
	<i>h</i> 13*	<i>j_s</i> 13*	<i>h</i> 14*	<i>j_s</i> 14*	<i>h</i> 15*	<i>j_s</i> 15*	<i>h</i> 16*	<i>j_s</i> 16*	<i>h</i> 17*	<i>j_s</i> 17*
	Предельные отклонения, мкм									
От 1 до 3	0	+70	0	+125	0	+200	0	+300	0	+500
	-140	-70	-250	-125	-400	-200	-600	-300	-1000	-500
Свыше 3 до 6	0	+90	0	+150	0	+240	0	+375	0	+600
	-180	-90	-300	-150	-480	-240	-750	-375	-1200	-600
Свыше 6 до 10	0	+110	0	+180	0	+290	0	+450	0	+750
	-220	-110	-360	-180	-580	-290	-900	-450	-1500	-750
Свыше 10 до 14	0	+135	0	+215	0	+350	0	+550	0	+900
Свыше 14 до 18	-270	-135	-430	-215	-700	-350	-1100	-550	-1800	-900
Свыше 18 до 24	0	+165	0	+260	0	+420	0	+650	0	+1050
Свыше 24 до 30	-330	-166	-520	-260	-840	-420	-1300	-650	-2100	-1050
Свыше 30 до 40	0	+195	0	+310	0	+500	0	+800	0	+1250
Свыше 40 до 50	-390	-195	-620	-310	-1000	-500	-1600	-800	-2500	-1250
Свыше 50 до 65	0	+230	0	+370	0	+600	0	+950	0	+1500
Свыше 65 до 80	-460	-230	-740	-370	-1200	-600	-1900	-950	-3000	-1500
Свыше 80 до 100	0	+270	0	+435	0	+700	0	+1100	0	+1750
Свыше 100 до 120	-540	-270	-870	-435	-1400	-700	-2200	-1100	-3500	-1750
Свыше 120 до 140	0	+315	0	+500	0	+800	0	+1250	0	+2000
Свыше 140 до 160	-630	-315	-1000	-500	-1600	-800	-2500	-1250	-4000	-2000
Свыше 160 до 180	0	+360	0	+575	0	+925	0	+1450	0	+2300
Свыше 180 до 200	-720	-360	-1150	-575	-1850	-925	-2900	-1450	-4600	-2300
Свыше 200 до 225	0	+405	0	+650	0	+1050	0	+1600	0	+2600
Свыше 225 до 250	-810	-405	-1300	-650	-2100	-1050	-3200	-1600	-5200	-2600
Свыше 250 до 280	0	+445	0	+700	0	+1150	0	+1800	0	+2850
Свыше 280 до 315	-890	-445	-1400	-700	-2300	-1150	-3600	-1800	-5700	-2850
Свыше 315 до 355	0	+485	0	+775	0	+1250	0	+2000	0	+3150
Свыше 355 до 400	-970	-485	-1550	-776	-2500	-1250	-4000	-2000	-6300	-3150
Свыше 400 до 450	0	+485	0	+775	0	+1250	0	+2000	0	+3150
Свыше 450 до 500	-970	-485	-1550	-776	-2500	-1250	-4000	-2000	-6300	-3150

Квалитеты 5 и 6

Интервал размеров, мм	Поля допусков												
	G5	H5	J _s 5	K5	M5	N5	G6	H6	J _s 6	K6	M6	N6	P6
	Предельные отклонения, мкм												
От 1 до 3	+6 +2	+4 0	+2,0 -2,0	0 -4	-2 -6	-4 -8	+8 +2	+6 0	+3,0 -3,0	0 -5	-2 -8	-4 -10	-6 -12
Свыше 3 до 6	+9 +4	+5 0	+2,5 -2,5	0 -5	-3 -8	-7 -12	+12 +4	+8 0	+4,0 -4,0	+2 -6	-1 -9	-5 -13	-9 -17
Свыше 6 до 10	+11 +5	+6 0	+3,0 -3,0	+1 -5	-4 -10	-8 -14	+14 +5	+9 0	+4,5 -4,5	+2 -7	-3 -12	-7 -16	-12 -21
Свыше 10 до 14	+14 +6	+8 0	+4,0 -4,0	+2 -6	-4 -12	-9 -17	+17 +6	+11 0	+5,5 -5,5	+2 -9	-4 -15	-9 -20	-15 -26
Свыше 14 до 18													
Свыше 18 до 24	+16 +7	+9 0	+4,5 -4,5	+1 -8	-5 -14	-12 -21	+20 +7	+13 0	+6,5 -6,5	+2 -11	-4 -17	-11 -24	-18 -31
Свыше 24 до 30													
Свыше 30 до 40	+20 +9	+11 0	+5,5 -5,5	+2 -9	-5 -16	-13 -24	+25 +9	+16 0	+8,0 -8,0	+3 -13	-4 -20	-12 -28	-21 -37
Свыше 40 до 50													
Свыше 50 до 65	+23 +10	+13 0	+6,5 -6,5	+3 -10	-6 -19	-15 -28	+29 +10	+19 0	+9,5 -9,5	+4 -15	-5 -24	-14 -33	-26 -45
Свыше 65 до 80													
Свыше 80 до 100	+27 +12	+15 0	+7,5 -7,5	+2 -13	-8 -23	-18 -33	+34 +12	+22 0	+11,0 -11,0	+4 -18	-6 -28	-16 -38	-30 -52
Свыше 100 до 120													
Свыше 120 до 140													
Свыше 140 до 160	+32 +14	+18 0	+9,0 -9,0	+3 -15	-9 -27	-21 -39	+39 +14	+25 0	+12,5 -12,5	+4 -21	-8 -33	-20 -45	-36 -61
Свыше 160 до 180													
Свыше 180 до 200													
Свыше 200 до 225	+35 +15	+20 0	+10,0 -10,0	+2 -18	-11 -31	-25 -45	+44 +15	+29 0	+14,5 -14,5	+5 -24	-8 -37	-22 -51	-41 -70
Свыше 225 до 250													
Свыше 250 до 280	+40 +17	+23 0	+11,5 -11,5	+3 -20	-13 -36	-27 -50	+49 +17	+32 0	+16,0 -16,0	+5 -27	-9 -41	-25 -57	-47 -79
Свыше 280 до 315													
Свыше 315 до 355	+43 +18	+25 0	+12,5 -12,5	+3 -22	-14 -39	-30 -55	+54 +18	+36 0	+18,0 -18,0	+7 -29	-10 -46	-26 -62	-51 -87
Свыше 355 до 400													
Свыше 400 до 450	+47 +20	+27 0	+13,5 -13,5	+2 -25	-16 -43	-33 -60	+60 +20	+40 0	+20,0 -20,0	+8 -32	-10 -50	-27 -67	-55 -95
Свыше 450 до 500													

Квалитет 7

Интервал размеров, мм	Поля допусков										
	<i>F7</i>	<i>G7</i>	<i>H7</i>	<i>J_s7</i>	<i>K7</i>	<i>M7</i>	<i>N7</i>	<i>P7</i>	<i>R7</i>	<i>S7</i>	<i>T7</i>
	Предельные отклонения, мкм										
От 1 до 3	+16 +6	+12 +2	+10 0	+5 -5	0 -10	-2 -12	-4 -14	-6 -16	-10 -20	-14 -24	-
Свыше 3 до 6	+22 +10	+16 +4	+12 0	+6 -6	+3 -9	0 -12	-4 -16	-8 -20	-11 -23	-15 -27	-
Свыше 6 до 10	+28 +13	+20 +5	+15 0	+7 -7	+5 -10	0 -15	-4 -19	-9 -24	-13 -28	-17 -32	-
Свыше 10 до 14	+34	+24	+18	+9	+6	0	-5	-11	-16	-21	-
Свыше 14 до 18	+16	+6	0	-9	-12	-18	-23	-29	-34	-39	
Свыше 18 до 24	+41	+28	+21	+10	+6	0	-7	-14	-20	-27	-
Свыше 24 до 30	+20	+7	0	-10	-15	-21	-28	-35	-41	-48	-33 -54
Свыше 30 до 40	+50	+34	+25	+12	+7	0	-8	-17	-25	-34	-39 -64
Свыше 40 до 50	+25	+9	0	-12	-18	-25	-33	-42	-50	-59	-45 -70
Свыше 50 до 65	+60	+40	+30	+15	+9	0	-9	-21	-30	-42	-55 -85
Свыше 65 до 80	+30	+10	0	-15	-21	-30	-39	-51	-62	-78	-64 -94
Свыше 80 до 100	+71	+47	+35	+17	+10	0	-10	-24	-38	-58	-78 -113
Свыше 100 до 120	+36	+12	0	-17	-25	-35	-45	-59	-73	-93	-41 -91 -126
Свыше 120 до 140									-48	-77	-107 -147
Свыше 140 до 160	+83 +43	+54 +14	+40 0	+20 -20	+12 -28	0 -40	-12 -52	-28 -68	-88	-117	-119 -159
Свыше 160 до 180									-53	-93	-131 -171
Свыше 180 до 200									-60	-105	-149 -195
Свыше 200 до 225	+96 +50	+61 +15	+46 0	+23 -23	+13 -33	0 -46	-14 -60	-33 -79	-106	-151	-163 -209
Свыше 225 до 250									-67	-123	-179 -225
Свыше 250 до 280	+108	+69	+52	+26	+16	0	-14	-36	-74	-138	-198 -250
Свыше 280 до 315	+56	+17	0	-26	-36	-52	-66	-88	-126	-190	-220 -272
Свыше 315 до 355	+119	+75	+57	+28	+17	0	-16	-41	-87	-169	-247 -304
Свыше 355 до 400	+62	+18	0	-28	-40	-57	-73	-98	-144	-226	-304 -273 -330
Свыше 400 до 450	+131	+83	+63	+31	+18	0	-17	-45	-103	-209	-307 -370
Свыше 450 до 500	+68	+20	0	-31	-45	-63	-80	-108	-166	-272	-370 -337 -400

Квалитеты 8 и 9

Интервал размеров, мм	Поля допусков													
	D8	E8	F8	H8	J _s 8	K8	M8	N8	U8	D9	E9	F9	H9	J _s 9*
	Предельные отклонения, мкм													
От 1 до 3	+34	+28	+20	+14	+7	0	-	-4	-18	+45	+39	+31	+25	+12
	+20	+14	+6	0	-7	-14	-	-18	-32	+20	+14	+6	0	-12
Свыше 3 до 6	+48	+38	+28	+18	+9	+5	+2	-2	-23	+60	+50	+40	+30	+15
	+30	+20	+10	0	-9	-13	-16	-20	-41	+30	+20	+10	0	-15
Свыше 6 до 10	+62	+47	+35	+22	+11	+6	+1	-3	-28	+76	+61	+49	+36	+18
	+40	+25	+13	0	-11	-16	-21	-25	-50	+40	+25	+13	0	-18
Свыше 10 до 14	+77	+59	+43	+27	+13	+8	+2	-3	-33	+93	+75	+59	+43	+21
Свыше 14 до 18	+50	+32	+16	0	-13	-19	-25	-30	-60	+50	+32	+16	0	-21
Свыше 18 до 24	+98	+73	+53	+33	+16	+10	+4	-3	-41	+117	+92	+72	+52	+26
Свыше 24 до 30	+65	+40	+20	0	-16	-23	-29	-36	-74	+65	+40	+20	0	-26
									-48					
Свыше 30 до 40	+119	+89	+54	+39	+19	+12	+5	-3	-81	+142	+112	+87	+62	+31
Свыше 40 до 50	+80	+50	+25	0	-19	-27	-34	-42	-60	+80	+50	+25	0	-31
									-99					
Свыше 50 до 65	+146	+106	+76	+46	+23	+14	+5	-4	-70	+174	+134	+104	+74	+37
Свыше 65 до 80	+100	+60	+30	0	-23	-32	-41	-50	-99	+100	+60	+30	0	-37
									-102					
Свыше 80 до 100	+174	+126	+90	+54	+27	+16	+6	-4	-148	+207	+159	+126	+87	+43
Свыше 100 до 120	+120	+72	+36	0	-27	-38	-48	-58	-124	+120	+72	+36	0	-43
									-178					
Свыше 120 до 140									-144					
Свыше 140 до 160	+208	+148	+106	+63	+31	+20	+8	-4	-198	+245	+185	+143	+100	+50
	+145	+85	+43	0	-31	-43	-55	-67	-170	+145	+85	+43	0	-50
Свыше 160 до 180									-233					
Свыше 180 до 200									-190					
									-253					
Свыше 200 до 225	+242	+172	+122	+72	+36	+22	+9	-5	-210	+285	+215	+165	+115	+57
Свыше 225 до 250	+170	+100	+50	0	-36	-50	-63	-77	-273	+170	+100	+50	0	-57
									-308					
Свыше 250 до 280	+271	+191	+137	+81	+40	+25	+9	-5	-330	+320	+240	+186	+130	+65
Свыше 280 до 315	+190	+110	+56	0	-40	-56	-72	-86	-284	+190	+110	+56	0	-65
									-356					
Свыше 315 до 355	+299	+214	+151	+89	+44	+28	+11	-5	-315	+350	+265	+202	+140	+70
Свыше 355 до 400	+210	+125	+62	0	-44	-61	-78	-94	-396	+210	+125	+62	0	-70
									-479					
Свыше 400 до 450	+327	+232	+165	+97	+48	+29	+11	-6	-431	+385	+290	+223	+155	+77
Свыше 450 до 500	+230	+135	+68	0	-48	-68	-86	-103	-390	+230	+135	+68	0	-77
									-587					

Квалитеты от 10 до 12

Интервал размеров, мм	Поля допусков											
	D10	H10	J _s 10	A11	B11	C11	D11	H11	J _s 11*	B12	H12	J _s 12*
	Предельные отклонения, мкм											
От 1 до 3	+60 +20	+40 0	+20 -20	+330 +270	+200 +140	+120 +60	+80 +20	+60 0	+30 -30	+240 +140	+100 0	+50 -50
Свыше 3 до 6	+78 +30	+48 0	+24 -24	+345 +270	+215 +140	+145 +70	+105 +30	+75 0	+37 -37	+260 +140	+120 0	+60 -60
Свыше 6 до 10	+98 +40	+58 0	+29 -29	+370 +280	+240 +150	+170 +80	+130 +40	+90 0	+45 -45	+300 +150	+150 0	+75 -75
Свыше 10 до 14	+120	+70	+35	+400	+260	+205	+160	+110	+55	+330	+180	+90
Свыше 14 до 18	+50	0	-35	+290	+150	+95	+50	0	-55	+150	0	-90
Свыше 18 до 24	+149	+84	+42	+430	+290	+240	+195	+130	+65	+370	+210	+105
Свыше 24 до 30	+65	0	-42	+300	+160	+110	+65	0	-65	+160	0	-105
Свыше 30 до 40	+180	+100	+50	+470 +310	+330 +170	+280 +120	+240	+160	+80	+420 +170	+250	+125
Свыше 40 до 50	+80	0	-50	+480 +320	+340 +180	+290 +130	+80	0	-80	+430 +180	0	-125
Свыше 50 до 65	+220	+120	+60	+530 +340	+380 +190	+330 +140	+290	+190	+95	+490 +190	+300	+150
Свыше 65 до 80	+100	0	-60	+550 +360	+390 +200	+340 +150	+100	0	-95	+500 +200	0	-150
Свыше 80 до 100	+260	+140	+70	+600 +380	+440 +220	+390 +170	+340	+220	+110	+570 +220	+350	+175
Свыше 100 до 120	+120	0	-70	+630 +410	+460 +240	+400 +180	+120	0	-110	+590 +240	0	-175
Свыше 120 до 140				+710 +460	+510 +260	+450 +200				+660 +260		
Свыше 140 до 160	+305 +145	+160 0	+80 -80	+770 +520	+530 +280	+460 +210	+395 +145	+250 0	+125 -125	+680 +280	+400 0	+200 -200
Свыше 160 до 180				+830 +580	+560 +310	+480 +230				+710 +310		
Свыше 180 до 200				+950 +660	+630 +340	+530 +240				+800 +340		
Свыше 200 до 225	+355 +170	+185 0	+92 -92	+1030 +740	+670 +380	+550 +260	+460 +170	+290 0	+145 -145	+840 +380	+460 0	+230 -230
Свыше 225 до 250				+1110 +820	+710 +420	+570 +280				+880 +420		
Свыше 250 до 280	+400	+210	+105	+1240 +920	+800 +480	+620 +300	+510	+320	+160	+1000 +480	+520	+260
Свыше 280 до 315	+190	0	-105	+1370 +1050	+860 +540	+650 +330	+190	0	-160	+1060 +540	0	-260
Свыше 315 до 355	+440	+230	+115	+1560 +1200	+960 +600	+720 +360	+570	+360	+180	+1170 +600	+570	+285
Свыше 355 до 400	+210	0	-115	+1710 +1350	+1040 +680	+760 +400	+210	0	-180	+1250 +680	0	-285
Свыше 400 до 450	+480	+250	+125	+1900 +1500	+1160 +760	+840 +440	+630	+400	+200	+1390 +760	+630	+315
Свыше 450 до 500	+230	0	-125	+2050 +1650	+1240 +840	+880 +480	+230	0	-200	+1470 +840	0	-315

Квалитеты от 13 до 17

Интервал размеров, мм	Поля допусков									
	<i>H</i> 13*	<i>J</i> _s 13*	<i>H</i> 14*	<i>J</i> _s 14*	<i>H</i> 15*	<i>J</i> _s 15*	<i>H</i> 16*	<i>J</i> _s 16*	<i>H</i> 17*	<i>J</i> _s 17*
	Предельные отклонения, мкм									
От 1 до 3	+140 0	+70 -70	+250 0	+125 -125	+400 0	+200 -200	+600 0	+300 -300	+1000 0	+500 -500
Свыше 3 до 6	+180 0	+90 -90	+300 0	+150 -150	+480 0	+240 -240	+750 0	+375 -375	+1200 0	+600 -600
Свыше 6 до 10	+220 0	+110 -110	+360 0	+180 -180	+580 0	+290 -290	+900 0	+450 -450	+1500 0	+750 -750
Свыше 10 до 14	+270 0	+135 -135	+430 0	+215 -215	+700 0	+350 -350	+1100 0	+550 -550	+1800 0	+900 -900
Свыше 14 до 18										
Свыше 18 до 24	+330 0	+165 -166	+520 0	+260 -260	+840 0	+420 -420	+1300 0	+650 -650	+2100 0	+1050 -1050
Свыше 24 до 30										
Свыше 30 до 40	+390 0	+195 -195	+620 0	+310 -310	+1000 0	+500 -500	+1600 0	+800 -800	+2500 0	+1250 -1250
Свыше 40 до 50										
Свыше 50 до 65	+460 0	+230 -230	+740 0	+370 -370	+1200 0	+600 -600	+1900 0	+950 -950	+3000 0	+1500 -1500
Свыше 65 до 80										
Свыше 80 до 100	+540 0	+270 -270	+870 0	+435 -435	+1400 0	+700 -700	+2200 0	+1100 -1100	+3500 0	+1750 -1750
Свыше 100 до 120										
Свыше 120 до 140										
Свыше 140 до 160	+630 0	+315 -315	+1000 0	+500 -500	+1600 0	+800 -800	+2500 0	+1250 -1250	+4000 0	+2000 -2000
Свыше 160 до 180										
Свыше 180 до 200										
Свыше 200 до 225	+720 0	+360 -360	+1150 0	+575 -575	+1850 0	+925 -925	+2900 0	+1450 -1450	+4600 0	+2300 -2300
Свыше 225 до 250										
Свыше 250 до 280	+810 0	+405 -405	+1300 0	+650 -650	+2100 0	+1050 -1050	+3200 0	+1600 -1600	+5200 0	+2600 -2600
Свыше 280 до 315										
Свыше 315 до 355	+890 0	+445 -445	+1400 0	+700 -700	+2300 0	+1150 -1150	+3600 0	+1800 -1800	+5700 0	+2850 -2850
Свыше 355 до 400										
Свыше 400 до 450	+970 0	+485 -485	+1550 0	+775 -776	+2500 0	+1250 -1250	+4000 0	+2000 -2000	+6300 0	+3150 -3150
Свыше 450 до 500										



УТВЕРЖДАЮ
Д.о. профессора по учебно-методической
работе В. В. Зубов

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ**

Б1.В.09.06 СРЕДСТВА ИЗМЕРЕНИЙ

Специальность -

21.05.04 Горное дело

Специализация -

Мехатроника и робототехника промышленных комплексов

Одобрена на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой

Волков Е.Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического

(название факультета)

Председатель

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

1. Общие положения

Курсовая работа – одна из важнейших форм самостоятельной работы студентов, способствующая углубленному изучению курса «Технические измерения и приборы», позволяющая исследовать приобретенные знания, совершенствовать навыки литературного изложения своих мыслей с использованием общенаучной терминологии.

В ходе подготовки и написания курсовой работы студенты приобретают навыки самостоятельного изучения определенной проблемы и умения самостоятельно делать обобщения.

Работа студентов над курсовой работой проходит следующие этапы:

- выбор темы исследования;
- составление библиографии по рассматриваемой проблеме;
- изучение учебной и специальной литературы, нормативных актов, статистических данных, общественно-политической практики;
- составление плана работы;
- написание текста работы и ее оформление с последующим представлением для рецензирования на кафедру.

1.1 Выбор темы курсовой работы

Студент знакомится с тематикой курсовых работ, содержащейся в данных методических рекомендациях, и выбирает заинтересовавшую его тему. При выборе темы рекомендуется исходить из интереса к той или иной проблеме, уже накопленного жизненного опыта. Необходимо также учитывать, что узкая, специальная тема потребует глубокого научного исследования проблемы, анализа источников различного вида.

При этом работа над одной темой нескольких обучаемых не допускается.

С разрешения руководства кафедры студент может выбрать тему, которая не значится в указанном перечне.

Сведения об избранной теме подаются на кафедру в установленные сроки.

Если в отведенный период времени обучаемым тема не избрана, она может быть определена преподавателем, ведущим занятие.

2. Примерная тематика курсовых работ

1. Приборы для измерения температуры.
2. Приборы для измерения давления.
3. Принцип работы пирометры излучения.
4. Измерение переменных и постоянных напряжений электронным

вольтметром.

5. Изучение принципа действия электронно-лучевого осциллографа.
6. Измерение частоты переменного тока и коэффициента мощности с помощью электронного осциллографа и частотомера.
7. Измерение температуры бесконтактным методом.
8. Методы и средства измерения переменных токов.
9. Принцип работы приборы электродинамической системы.
10. Принцип работы приборы Ферродинамические приборы
11. Автоматические измерительные приборы с дифференциально трансформаторными и ферро-динамическими преобразователями.
12. Принцип работы термоэлектрические преобразователи.
13. Тензорезистивные манометры. Схемы их подключения.
14. Выбор средств измерения температуры на узле учета нефти.
15. Методы и приборы для измерения вязкости веществ.
16. Принцип работы биполярного транзистора.
17. Принцип работы полярного транзистора.
18. Бесконтактные методы и приборы для измерения расхода веществ (электромагнитные, ультразвуковые).
19. Выбор средства измерения давления на входе установки осушки газа.
20. Методы и приборы для измерения физических свойств веществ.

Плотномеры.

21. Выбор средства измерения уровня в электродегидраторе на установке подготовки нефти.

22. Пирометры излучения.

23. Выбор средства измерения давления перегретого пара котельного агрегата.

24. Термоэлектрические преобразователи.

25. Выбор средства измерения температуры верха ректификационной колонны.

26. Принцип работы магнитоэлектрические приборы.

27. Преобразователь э.д.с. термопары и сопротивления термометра в ток.

28. Выбор средства измерения давления верха ректификационной колонны.

29. Принцип работы Пьезоэлектрические датчики.

30. Исследование датчиков углового перемещения.

31. Исследование датчиков частоты вращения.

32. Выбор средства измерения давления перед низкотемпературным сепаратором.

33. Одинарный индуктивный датчики.

34. Выбор средства измерения и сигнализации температуры в дренажной

емкости.

35. Принцип работы магнитоэлектрические приборы.

36. Индукционные преобразователи.

37. Выбор средства измерения давления в барабане котла.

38. Измерение мощностей и энергии.

39. Выбор средства измерения температуры в низкотемпературном сепараторе.

40. Классификация и принципы построения цифровые измерительные приборы (ЦИП).

41. Цифровые измерительные приборы основных видов и назначений.

42. Выбор средства измерения температуры пара для испарителя.

43. Измерительные мосты постоянного и переменного токов.

44. Цифровые измерительные приборы (ЦИП).

45. Выбор средства измерения и сигнализации температуры в дренажной емкости.

46. Измерения и регистрация изменяющихся во времени электрических величин. Электронно-лучевой осциллограф.

47. Метод измерения индуктивности, добротности, емкости и тангенса угла потерь.

3. Методические рекомендации по выполнению курсовой работы

3.1 ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

В настоящее время чрезвычайно расширяется ассортимент измерительных средств общего и специального назначения (датчиков, приборов и измерительных систем), устройство и применение которых осваивается студентами.

При курсовом проектировании *студент должен*:

- развить навыки самостоятельной творческой работы;
- использовать элементы научно-исследовательского подхода при решении инженерных задач.

Выполнение курсового проекта должно способствовать закреплению, углублению и обобщению знаний, получаемых студентами в процессе изучения настоящего курса: знаний принципа действия; устройства основных узлов и приборов в целом; особенности характеристик и прикладные вопросы используемых измерительных устройств; общие соображения по выбору конкретных средств измерения (СИ), имея в виду их особенности, качество, возможность сочетания и обслуживания с другими устройствами для измерений и автоматизации; эксплуатационные характеристики при использовании СИ; правила

приемки и поверки СИ. Кроме того, в процессе работы усваиваются навыки применения этих знаний к комплексным разработкам конкретных инженерных проблем, подготавливая тем самым студентов к выполнению дипломного проектирования и к текущей самостоятельной работе на производстве. Все это преследует цель – развитие способностей к самостоятельному инженерному творчеству.

Наряду с этим выполнение курсового проекта должно научить студентов пользоваться всякого рода справочной литературой, каталогами и типовыми проектными разработками отечественных и зарубежных фирм. В ходе курсового проектирования студентам необходимо освоить практику проведения расчетов, применяемых для конкретных СИ, при оценке погрешностей результатов измерений в реальных условиях эксплуатации, а также научиться технически грамотно составлять пояснительные записки, включая составление спецификации по контрольно-измерительной аппаратуре при разработке принципиальных схем технологического контроля, измерения и автоматизации для рассматриваемого технологического процесса и обоснование выбора конкретных элементов и комплектов измерительной техники для систем контроля и управления.

3.2 ТЕМАТИКА КУРСОВОГО ПРОЕКТА

Так как от правильного определения тематики курсовых проектов зависят научность, современность и реальность разрабатываемого проекта, то ее выбору должно уделяться большое внимание. Последнее тем более справедливо, что задание на курсовое проектирование обязательно должно быть индивидуальным и согласованно с интересами, способностями и направлением работы каждого студента без снижения требований.

3.3 СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТА

Курсовой проект состоит из *графической части* и *пояснительной записки* к ней.

Графическая часть включает в себя:

- функциональную схему автоматизации (в части выбора средств измерения) технологического участка на одном листе формата А4;
- принципиальную механическую или электрическую схему выделенного из схемы измерительного устройства на одном листе формата А4;
- установочный чертеж выделенного из схемы измерительного датчика или прибора или монтажный чертеж установки прибора на щите на 1 листе формата А4.

Требования к оформлению графической части

Графическая часть должна быть выполнена в соответствии с требованиями стандартов на обозначения основных величин и условное изображение приборов в схемах автоматизации производственных процессов.

Пояснительная записка содержит:

- титульный лист (Приложение 1);
- задание на разработку проекта, подписанное руководителем проектирования и студентом, выполняющим проект (Приложение 2);
- *первую часть*, посвященную краткому описанию технологических процессов, их характеристик и параметров, а также особенности эксплуатационных характеристик используемого оборудования для выбранного участка (цеха) производства, подробной спецификации на измерительные устройства, используемые в схеме управления или контроля с подробным описанием обоснования выбора основных 5-6 измерительных комплектов;
- *вторую часть*, состоящую из расчета метрологических характеристик (мх) измерительного канала информационно-измерительной системы для определения одного из выбранных СИ;
- *третью часть*, - реферат, содержащий описание СИ (принцип действия, устройство, характеристики, область и особенности применения);
- библиографический список;
- содержание.

Требования к оформлению пояснительной записки

Пояснительная записка должна быть напечатана и изложена технически грамотно, четко и сжато. Решения, приводимые в ней, должны иметь обоснования путем сравнения имеющихся и возможных вариантов и выбором лучших из них под углом зрения метрологического обеспечения систем контроля, измерения и управления, унификации, простоты, дешевизны, а также удобства эксплуатации, ремонта и обеспечения техники безопасности в работе. Расчеты должны сопровождаться необходимыми схемами, графиками и эскизами. Располагать расчеты в записке следует в той последовательности, в которой они выполнялись. Все расчетные формулы, приведенные в записке, следует вначале написать в буквенном выражении, а затем в том же порядке, в каком даны буквы, в формулы подставить их численные значения и полученные результаты. Входящие в формулы буквенные обозначения, за исключением общепринятых, должны иметь пояснения. Во всех расчетах должна применяться Международная система единиц.

Приводимые расчеты должны быть обоснованы ссылкой на соответствующую литературу и источники, список которых помещается в конце пояснительной записки.

3.4 УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

Первая часть курсового проекта включает в себя:

- краткое описание технологического процесса; при

этом необходимо:

- дать краткое описание места данного технологического процесса в общем процессе выработки продукции;
- по возможности привести стехиометрические уравнения реакций, происходящие в данном аппарате, реакторе, цехе и дать краткое описание технологического процесса;
- привести технологический регламент работы оборудования данного участка производства;
- дать краткую техническую характеристику каждой единице оборудования, принадлежащей рассматриваемому объекту, аппарату, цеху;
- указать особое условие эксплуатации СИ с указанием температуры, статического давления, специфические характеристики измеряемой среды, агрессивность окружающей среды и ее параметры;

- выбор параметров, подлежащих контролю и управлению;

при этом необходимо, помимо перечня параметров, определить погрешности, допускаемые при контроле или измерении того или иного технологического параметра, руководствуясь основными условиями ведения технологического процесса и постановкой измерительной задачи.

- таблица контролируемых параметров;

в таблице необходимо указать специфические характеристики измеряемой среды и условия работы СИ, а также средние (номинальные) значения измеряемого параметра и необходимую погрешность измерения параметра в абсолютных или относительных значениях.

Форма таблицы представлена табл. 1. В таблицу заносятся все параметры, подлежащие контролю или измерению. Условия работы СИ и места их установки уточняются во время ознакомительной практики, либо используются литературные данные;

Контролируемые параметры технологического процесса

Измеряемый параметр		особенности измеряемого параметра, характеристика измеряемой среды	характеристика окружающей среды	требования к погрешности	примечание
наименование	пределы измерения				
1	2	3	4	5	6

- обоснование выбора измерительных комплектов;

подробное обоснование выбора измерительного комплекта должно производиться в соответствии с:

- постановкой измерительной задачи, при этом указывается, какой технологический параметр, где и в каких эксплуатационных условиях, при каких дополнительных требованиях должен быть измерен с заданными метрологическими характеристиками, удовлетворяющими требованиями технологического контроля производственного процесса;
- перечислением возможных вариантов принципиального решения измерительной задачи, при этом необходимо перечислить все измерительные устройства, которые могут решать поставленную задачу с точки зрения принципа действия;
- выбором вариантов измерительных комплектов, реализующих метрологические требования: здесь проводится расчет основных метрологических характеристик измерительных комплектов для измерения данного информативного параметра в соответствии с имеющимися метрологическими характеристиками датчиков и приборов и правилами их нормирования и суммирования;
- выбором окончательного варианта измерительного комплекта, _ при выборе окончательного варианта измерительной аппаратуры, если сравнение по метрологическим характеристикам равнозначно, должно отдаваться предпочтение:

а) современным средствам измерения;

б) комплектам ГСП, унифицированным по принципу действия с остальной аппаратурой, используемой на производстве;

в) наиболее простым, надежным и дешевым средствам измерения из сравниваемых вариантов;

- заполнение спецификации на контрольно-измерительные приборы;

порядок заполнения спецификации желателен следующий:

- 1) температура;
- 2) давление;
- 3) расход;
- 4) уровень;
- 5) физико-химические свойства;
- 6) специальные приборы.

После заполнения спецификации на функциональную схему автоматизации проставляются номера измерительных комплектов согласно номерам позиций спецификации.

Форма спецификации должна соответствовать табл. 2.

Таблица 2

Позиция	Наименование и техническая характеристика	Тип, марка	Завод-изготовитель	Количество

Образец выполнения курсовой работы

ПРИМЕР ВЫПОЛНЕНИЯ ПЕРВОЙ ОБЩЕЙ ЧАСТИ

КУРСОВОГО ПРОЕКТА

*Описание технологического процесса на примере процесс
сульфатной варки*

Варка сульфатной целлюлозы осуществляется периодическим способом в вертикальных стационарных котлах клепаной или сварной конструкции, стальной корпус которых изнутри защищен кислотоупорной футеровкой.

Котел загружают щепой из бункера, заливают варочным раствором и герметически закрывают. Затем содержимое котла нагревают паром до температуры 105-110°C (заварка) и выдерживают при этой температуре для завершения пропитки щепы кислотой. Далее нагрев продолжают до конечной температуры варки – до 128-155°C. Различают прямой и непрямой нагрев варочного котла. При прямом нагреве пар поступает непосредственно в котел и там конденсируется, при непрямом нагреве пар поступает в установленный рядом с котлом поверхностный подогреватель, через который непрерывно прогоняется циркуляционным насосом варочная жидкость. В последнем случае конденсат пара не смешивается с раствором. Рабочее давление в котле может составлять от 0,6 до 1 МПа.

В зависимости от условий продолжительность варки колеблется в широких пределах – примерно от 5 до 12 ч. В процессе варки из котла периодически производят сдувки, т.е. удаляют из верхней части котла водяной пар и сернистый газ в систему регенерации, где они смешиваются с сырой щелочью, образуя варочные щелока.

Ход варки контролируется варщиком по показанию приборов и по результатам анализа варочной жидкости. В конце варки иногда отбирают из котла пробы сваренной массы. Отработанная варочная жидкость носит название черного сульфатного щелока.

По окончании варки котел опоражнивают в сжезу или приемный резервуар способом выдувки или вымывки. При выдувке давление в котле снижают до 0,15-0,25 МПа, открывают шабер на выдувном трубопроводе и целлюлозная масса остаточным давлением выдувается вместе со щелоком в выдувной резервуар. Продолжительность выдувки 10-20 минут. При вымывке давление в котле снижают до 0,25-0,35 МПа и предварительно отбирают из котла сульфатный щелок, направляемый на регенерацию, то есть

восстановление. В котел, заполняя его до верху, подают воду или слабый оборотный щелок и еще некоторое время продолжают отбор крепкого щелока под гидравлическим давлением, затем выпускают массу с оставшимся слабым щелоком из котла в вымывной резервуар. Продолжительность всех операций при вымывке составляет 1,5-2 ч.

Выбор и анализ параметров, подлежащих контролю и управлению

Функциональная схема контроля, измерения и автоматизации процесса варки сульфатной целлюлозы приведена на рис. 1.

В процессе варки целлюлозы в котле наиболее важной величиной является **температура варочной жидкости**, так как она непосредственно влияет на пропитку древесины щелоком, время варки, выход и качество целлюлозы. Чрезмерная температура снижает потенциальный выход котла за счет физического и химического разрушения целлюлозы и ее частичного разложения на растворимые продукты, которые в дальнейшем удаляются из массы в процессе промывки. В это же время для обеспечения приемлемого качества целлюлозы температура в варочном котле должна быть достаточно высокой. Измерение температуры должно производиться с минимальной погрешностью дистанционно.

Давление газа, выделяющегося из щелока в процессе варки, непосредственно связано с крепостью щелока и скоростью проникновения последнего в щепу. Это давление должно быть достаточно высоким, чтобы обеспечить максимальную производительность котла, а для равномерной пропитки и получения продукции с минимальным содержанием серы

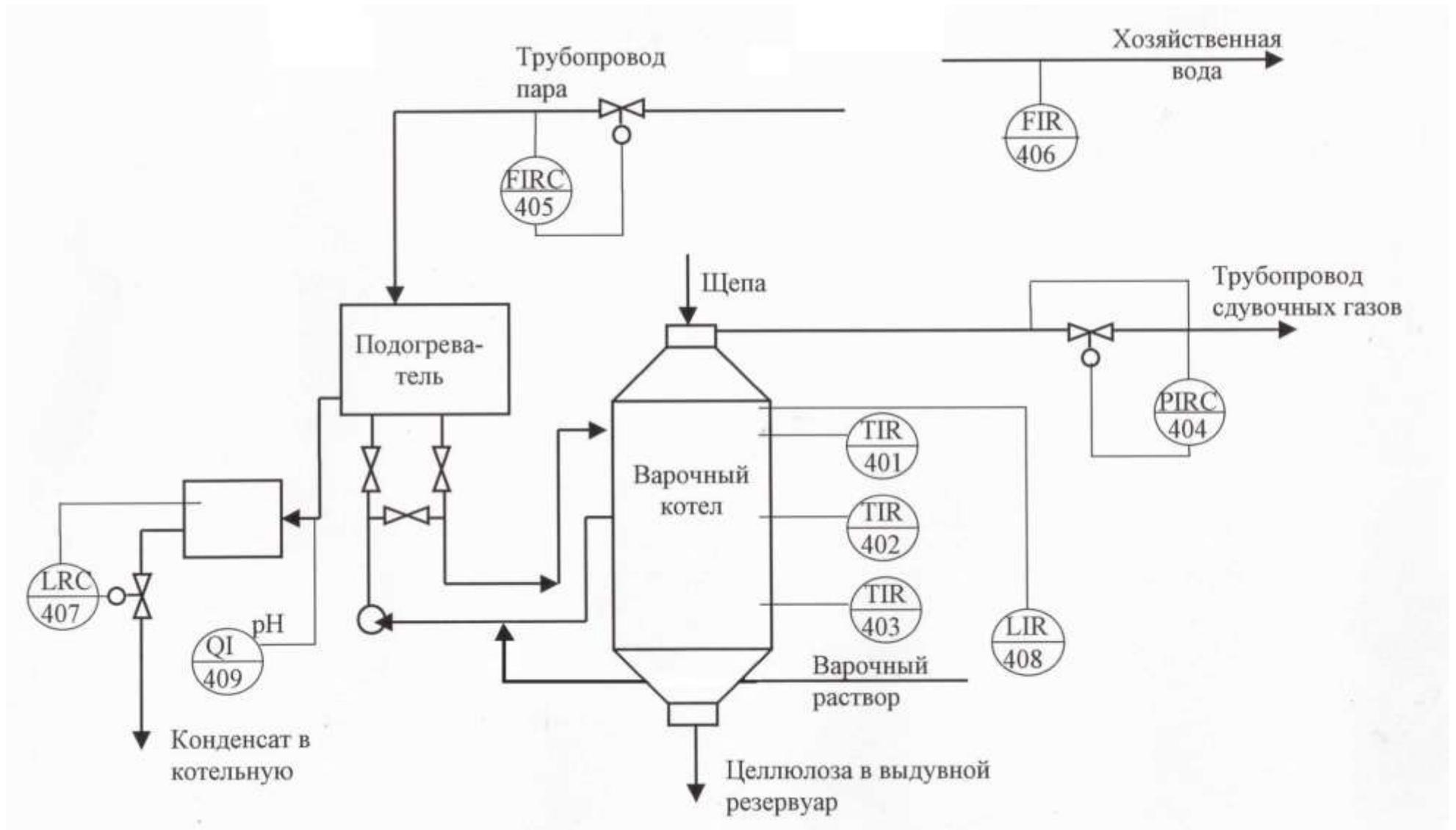


Рис. 1 Функциональная схема автоматизации процесса варки сульфатной целлюлозы

указанное давление необходимо стабилизировать с помощью системы автоматического управления. Управление предельным давлением сдувок имеет важное значение на протяжении всего варочного процесса, но особенно его значение возрастает во время заключительного периода сдувок газа в конце варки.

Необходимо осуществлять контроль **уровня щелока** в котле, он должен быть достаточным, чтобы накрывать щепу в начале варки, в то же время над щелоком должно оставаться некоторое пространство в верхней части варочного котла, позволяющее обеспечить максимальное извлечение газа и предотвратить «сырую» варку. Уровень в котле должен измеряться уровнемером с дистанционным представлением результатов измерений.

В подогревателе на линии конденсата пара необходимо контролировать **уровень конденсата и возможность попадания в конденсат варочной жидкости** в случае течи трубок. Для этого следует предусмотреть установку рН-метра с сигнальным устройством. Кроме того, важными контролируемыми параметрами являются температура, давление и расход пара, требуемого для нагревания содержимого варочного котла, а также температура, давление и расход хозяйственной воды.

Таблица контролируемых параметров

Все описанные в п. 5.2. контролируемые параметры заносятся в таблицу, указываются специфические характеристики измеряемой среды, номинальные значения измеряемого параметра и необходимую погрешность измерения.

Обоснование выбора измерительных комплексов

Рассмотрим пример обоснования выбора измерительных комплексов при измерении температуры в варочном котле.

Таблица 3

Контролируемые параметры процесса варки сульфатной целлюлозы

Измеряемый параметр		Место измерения параметра	Характеристика среды	Абсолютная погрешность	Примечание
наименование параметра	пределы измерения				
Температура варочной жидкости	0-200 °С	3 зоны варочного котла	Среда агрессивная, загазованная, T=200 °С	$\Delta \leq \pm 1^\circ\text{C}$	$\delta = \frac{1 * 100}{200} \pm 0,5\%$
Уровень варочной жидкости	13-18 мм	Варочный котел	Среда агрессивная, загазованная, T=200 °С	$\Delta = \pm 0,20\text{м}$	$\delta = \frac{0,2 * 100}{200} \pm 1\%$
Давление сдувочных газов	0-1МПа	Трубопровод сдувочных газов	Среда агрессивная, загазованная, T=200 °С	$\Delta = \pm 0,05$ МПа $\pm 0,01\text{МПа}$ $\pm 0,02\text{МПа}$	$\delta = \frac{0,05 * 100}{1} \pm 5\%$
Уровень конденсата	0-5м	Подогреватель	Среда неагрессивная p=0,5МПа T=200 °С	$\Delta = \pm 0,05\text{м}$	$\delta = \frac{0,05 * 100}{5} \pm 1\%$
pH	0-7 pH	Паропроводник подачи пара в подогреватель	-----	$\Delta = \pm 0,05\text{pH}$	$\delta = \frac{0,05 * 100}{7} \pm 0,7\%$
Температура пара	180 °С	---	---	$\Delta = \pm 1^\circ\text{C}$	$\delta = \frac{1 * 100}{180} \pm 0,5\%$
Давление пара	10 кг/см ²	---	P=1МПа	$\Delta = \pm 1$ кг/см ²	$\delta = \frac{0,1 * 100}{10} \pm 1\%$
Расход пара	0-200 м ³ /ч	---	Среда неагрессивная p=0,5МПа T=200°С	$\Delta = \pm 1\text{м}^3/\text{ч}$	$\delta = \frac{10 * 100}{200} \pm 0,5\%$
Температура воды	40 °С	Трубопровод хозяйственной воды	P=5кг/см ²	$\Delta = \pm 1^\circ\text{C}$	$\delta = \frac{1 * 100}{40} \pm 2,5\%$

1. Сформулируем измерительную задачу: требуется измерить температуру от 80 до 200°С в варочном котле при значительном давлении P=1,0 МПа. Требование к погрешности измерения температуры самые жесткие, т.е.

следует выбирать наиболее точные и быстродействующие средства измерения. Измеряемую величину следует определять с помощью показывающего прибора, регистрировать дистанционно и предусмотреть унифицированный выходной сигнал для использования в системе управления.

2. Перечислим средства измерения, которые могут определять указанную температуру дистанционно (по принципу действия) и контактно (иначе говоря, с наибольшей

чувствительностью).

К ним относятся:

- манометрические термометры (дистанция до 60 м);
- термометры электрического сопротивления (ТЭС) (проводниковые и полупроводниковые с измерительными унифицирующими преобразователями (для унификации измерительного сигнала) и приборами (для регистрации);
- термоэлектрические термометры с измерительными унифицирующими преобразователями (для унификации измерительного сигнала) и приборами (для регистрации).

Манометрический термометр характеризуется по точности приведенной погрешностью $\gamma = \pm(1,5-2,5)\%$, что не соответствует заданным требованиям, не говоря о динамических характеристиках.

Термоэлектрические термометры стандартные из благородных материалов имеют, как правило, погрешность невоспроизводимости $\Delta = \pm 0,01 \text{ мВ}$, а при температуре, равной 200°C , развивают ЭДС равную $E = 1,436 \text{ мВ}$. Тогда относительная

погрешность измерения составит:

$$\delta = \pm \frac{\Delta \cdot 100}{X} = \pm \frac{\Delta \cdot 100}{E} = \pm \frac{0,01 \cdot 100}{1,436} = \pm 0,7 \%$$

или по абсолютной величине температуры:

$$\Delta = \pm \frac{\delta \cdot T}{100} = \pm \frac{0,7 \cdot 200}{100} = \pm 1,4^\circ\text{C},$$

что не соответствует заданным требованиям к измерениям температуры.

Стандартные термоэлектрические термометры из неблагородных металлов оцениваются погрешностью невоспроизводимости около

$\Delta = \pm 0,2 \text{ мВ}$. Тогда относительная погрешность измерения заданной температуры для наиболее чувствительной термопары типа К(никель- хром/никель-алюминий) при

Составит $E_{(200\text{C})} = 14,66 \text{ мВ}$

а абсолютная погрешность равна:

$$\Delta = \frac{\delta \cdot T}{100} \pm \frac{1,36 \cdot 200}{100} = \pm 2,73^\circ\text{C}.$$

Сделаем вывод о непригодности термоэлектрических термометров для решения поставленной задачи, т.е. для измерения низких температур с малыми погрешностями.

ТЭС из платины (как наиболее стабильные и точные) выпускаются с допуском отклонения номинального значения сопротивления

$\Delta R \cdot 100 / R_0 = \pm 0,06\%$ (группа А) и поэтому являются датчиками температуры с

наименьшей погрешностью в диапазоне измеряемых температур от -260 до $+1100^\circ\text{C}$.

Выбор номинального значения сопротивления термопреобразователей при 0°C , т.е. R_0 в пределах от 1 до 2000 Ом должен учитывать то обстоятельство, что чем больше R_0 , тем больше абсолютная чувствительность ТЭС, которая в первом приближении:

$$S_{abc} = R_0 \cdot \alpha,$$

где α - температурный коэффициент электрического сопротивления материала ТЭС, в данном случае платины.

Предпочтительные значения R_0 следует выбирать: 100, 200, 500 или 1000 Ом с учетом незначительной массы провода при разумном его диаметре для исключения большой инерционности и обрывов провода соответственно.

Тогда выберем $R_0=500$ Ом. Погрешности измерения ТЭС можно определить из следующих соображений, выбрав соответствующую марку платинного датчика, а именно, ТСПУ 015, который используется для измерения температур в диапазоне $0-200^\circ\text{C}$. Он имеет следующие метрологические характеристики: основная приведенная погрешность

$\gamma_0 = \pm 0,5\%$, дополнительная погрешность от изменения напряжения питания

$\gamma_0 = \pm 0,1\%$; в диапазоне изменения температуры окружающей среды $T_{\text{ср}}=(-50 \div +70)^\circ\text{C}$

дополнительная температурная погрешность составляет

$$\Delta \gamma_T = \frac{\pm 0,2\%}{\pm 10^\circ\text{C}};$$

дополнительная погрешность от изменения сопротивления нагрузки $\gamma_{\text{ннн}} = \pm 0,1\%$; инерционность – 15с; допусковое давление среды

$P_{cp} \leq 16$ МПа (что соответствует требованиям измерительной задачи) Тогда

основная абсолютная погрешность составит:

$$\Delta_0 = \frac{\gamma_0 \cdot T_{max}}{100\%} = \frac{\pm 0,5\% \cdot 200^0 C}{100\%} = \pm 1^0 C$$

Дополнительная погрешность при изменении температуры внешней среды от $T_n = (20 \pm 5)^0 C$ до $T_{cp} = 40^0 C$ (температуры окружающей среды) составит:

$$\gamma_{T_{доп}} = \gamma_T \cdot (T_{cp} - T_{nmax}) = \gamma_T \cdot (40 - 25)^0 C = \pm \frac{0,2\%}{10^0 C} \cdot 15^0 C = \pm 0,3\%$$

Суммарная дополнительная погрешность ТЭС с учетом всех внешних влияющих факторов будет равна:

$$\gamma_{допТЭС} = \left(\sum_{i=1}^3 \gamma_i^2 \right)^{1/2} = (\gamma_n^2 + \gamma_{T_{доп}}^2 + \gamma_{R_{ннн}}^2)^{1/2} = \pm (0,1^2 + 0,3^2 + 0,1^2)^{1/2} \cong \pm 0,3\%$$

абсолютной форме:

$$\Delta_{допТЭС} = \frac{\gamma_{допТЭС} \cdot T_{max}}{100\%} = \pm \frac{0,3 \cdot 200}{100} = \pm 0,6 C$$

Тогда полная погрешность ТСА Δ_p в реальных условиях эксплуатации

составит:

$$\Delta_p = \Delta_0 + \Delta_{допТЭС} = \pm (1 + 0,6) \cdot C = \pm 1,6 C.$$

Для работы в комплексе с ТЭС используются (если необходимо устанавливать показывающие и (или) регистрирующие приборы):

-логометры магнитоэлектрической системы;

-неуравновешанные мосты постоянного и переменного тока с указателями неравновесия в виде логометров, милливольтметров или миллиамперметров (обычно магнитоэлектрической системы с выпрямителями, если используется мост переменного тока);

-пирометрические магнитоэлектрические узкопрофильные милливольтметры со встроенными предварительными преобразователями мостового типа, например, из серии АСК типов МВУ6 (последние со встроенными сигнализаторами, позиционными регуляторами и выходными преобразователями);

-миллиамперметры магнитоэлектрической системы в комплекте с блоками унифицированных стандартных преобразователей, имеющих выходной стандартный сигнал постоянного тока 4-20мА;

-автоматические электронные уравновешенные мосты постоянного тока, например, приборы типа КМС (самопишущие) и КПМ (показывающие).

Алгебраическая сумма абсолютных погрешностей датчика и моста дает максимально возможную абсолютную погрешность выбранного комплекта.

Геометрическая сумма абсолютных погрешностей как независимых случайных величин дает значение абсолютной погрешности выбранного комплекта для определения температуры:

$$\Delta = \sqrt{\Delta_o^2 + \Delta_M^2} = \sqrt{1,6^2 + 0,5^2} = \sqrt{2,56 + 0,25} = \pm \sqrt{0,25} = \pm 0,5^0 C$$

Таким образом, из возможных вариантов следует остановиться на измерительном комплекте платинового ТЭС, класса точности А и КСМ-4 класса точности 0,25 со временем установления показаний 20 с и со встроенными преобразователями унифицированных сигналов с $I_{\text{вых}}=4-20\text{мА}$ для использования в системе управления.

При выборе конкретных типов КСМ-4 и ТЭС следует учитывать условия работы в варочном цехе при изменении температуры варочного щелока (агрессивного вещества) в трубопроводе с учетом рабочего статистического давления $P=1\text{МПа}$.

Правила установки датчика температуры должны соответствовать следующим основным положениям, вытекающим из анализа погрешностей измерения температур датчиками через потери тепла:

- датчики температур следует погружать как можно глубже в измеряемую среду;
- датчик устанавливают в месте наибольшей скорости движения среды и против направления ее движения;
- трубопровод или объект исследования в месте установки датчика тщательно теплоизолирован, защищая головку датчика от воздействия температуры окружающей среды.

После выбора средств измерений их заносят в спецификацию как показано в примере табл. 4.

Таблица 4

Спецификация на СИ для варочного котла

Позиция	Наименование и техническая характеристика	Тип, марка	Завод – изготовитель	Количество
	Приборы и средства автоматизации			
TIR-401	Контроль температуры в верхней зоне варочного котла	ТСПУ 015	СКБ Термоприбор, Москва	1
	Температура 80-155°C			
РС-404	Регулирование давления сдувок в трубопроводе сдувочных газов. Давление 1 МПа			
РЕ-404-1	Отбор импульсов давления с разделительным устройством			1
РЕ-404-2	Измерительный преобразователь разности давлений. Выход 4...20мА. Приведенная погрешность $\gamma=\pm 0,5\%$	Метран 100Д	Теплоприбор, г. Челябинск	1
ФС-405	Регулирование расхода пара в трубопроводе. Расход 200 м ³ /ч			
FE-405-1	Сужающее устройство.	ДК-10-200	Теплоприбор, г. Челябинск	1
FT-405-2	Измерительный преобразователь разности давлений. Выход 4...20мА. Приведенная погрешность $\gamma=\pm 0,5\%$	Метран 100 ДД		1

Позиция	Наименование и техническая характеристика	Тип, марка	Завод – изготовитель	Количество
FIR-406	Контроль расхода хоз. воды F=180-200 м ³ /ч			
FIR-406-1	Сужающее устройство	ДК-10-200		1
FT-406-2	Измерит. преобразователь разности давлений. Выход 4...20мА. Привед. погрешность $\gamma=\pm 0,5\%$	Метран 100 ДД	Теплоприбор, г. Челябинск	1
LRC-407	Регулирование уровня конденсата в подогревателе. Уровень 5м.			
LE-407-1	Отбор импульсов давления: разделительное устройство	Метран 100 ДД	Теплоприбор, г. Челябинск	1
LT-407-2	Измерит. преобразователь разности давлений. Выход 4...20мА. Привед. погрешность $\gamma=\pm 0,5\%$	Метран 100 ДД	Теплоприбор, г. Челябинск	1
LIR-408	Контроль уровня варочной жидкости в варочном котле. Уровень 13-18м			
LE-408-1	Отбор импульсов давления: разделительное устройство	Метран 100 ДД	Теплоприбор, г. Челябинск	1
LT-408-2	Измерит. преобразователь разности давлений. Выход 4...20мА. Привед. погрешность $\gamma=\pm 0,5\%$	Метран 100 ДД	Теплоприбор, г. Челябинск	1
QIR-409	Контроль pH конденсата в подогревателе. pH=7			
QE-409-1	Стеклянный электрод; хлорсеребряный электрод	«Атлант» Мод. 2101	ООО «Артеко», Москва	1

ПРИМЕР ОРИЕНТИРОВОЧНОГО РАСЧЕТА МЕТРОЛОГИЧЕСКИХ ХАРАКТЕРИСТИК ИЗМЕРИТЕЛЬНОГО КАНАЛА

ПО МЕТРОЛОГИЧЕСКИМ ХАРАКТЕРИСТИКАМ ЕГО ЭЛЕМЕНТОВ

Ориентировочный расчет метрологических характеристик измерительного канала необходимый для основания выбора средств измерения, составляющих измерительный канал при их проектировании.

Так как при проектировании неизвестны условия эксплуатации, а также подробные характеристики средств измерения (СИ), то ориентировочный расчет метрологических характеристик (МХ) измерительного канала (ИК) ведется с рядом допущений:

1. СИ, входящие в состав ИК, являются линейными либо линеаризуемыми средствами автоматизации, так как, как правило, при контроле и управлении технологических параметров измерения осуществляется в незначительном диапазоне изменения физической величины, т.е. это допущение реализуется.

2. Погрешности СИ и результирующие погрешности каналов ИК представляют величины второго порядка малости по сравнению с параметрами соответствующих номинальных статических характеристик преобразования средств измерения в ИК.

3. Отклонения значений метрологических характеристик ИК, вызванные изменением погрешности под действием изменения внешних влияющих величин (ВВВ) в статических и динамических режимах малы по сравнению с самими номинальными значениями характеристик СИ.

4. Погрешности СИ ИК независимы друг от друга, т.е. коэффициенты их корреляции можно считать близкие к значениям, характеризующим независимые величины ($\rho=0\div 0,7$).

5. Оценки пределов допускаемых погрешностей назначают по ГОСТ 8.401-80 для всех СИ ИК. Сопротивление отдельных элементов канала согласованы между собой. Взаимное влияние между ИК не учитывается.

Поскольку сделанные допущения не всегда справедливы, то конкретные значения погрешности ИК в реальных условиях эксплуатации определяются практикой, а именно в период опытно-промышленной эксплуатации каналов после их монтажа и отладки, проводится поверка или калибровка ИК.

Структура измерительного канала представлена на рис. 2.



Д – датчик
 УП – унифицирующий преобразователь
 ДЛС – дистанционная линия связи
 МП – масштабирующий преобразователь
 АЦП – аналогово-цифровой преобразователь
 ПР – процессор
 Д – дисплей

Рис. 2. Структура измерительного канала

Для примера расчета МХ ИК рассмотрим канал измерения температуры в варочном котле, конкретная структура которого приведена на рис. 3.

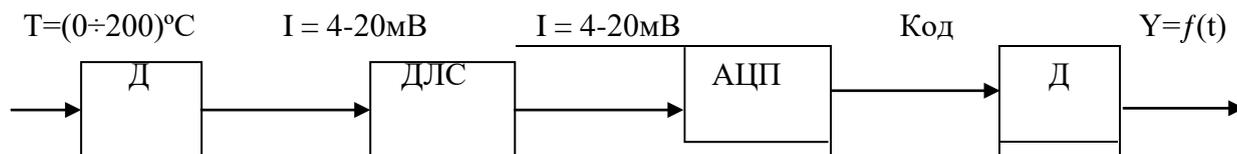


Рис. 3. Структурная схема канала измерения температуры в варочном

Исходные данные

1. ДТ – ТЭС тип ТСПУ 014. Предел измерения температуры (0-200)°С. Погрешности датчика: основная – $\gamma_u = \pm 0,5\%$ /R0=500 Ом/; дополнительные: по напряжению $\gamma_u = \pm 0,1\%$; по температуре для интервала температур $T_{окр.ср} = (-50 \div +70)^\circ\text{C}$ $\gamma_T = \pm 0,2\% / \pm 10^\circ\text{C}$; по изменению нагрузки следующего преобразователя $\gamma_{R \text{ нагр}} = \pm 0,1\%$.

2. ДЛС – витая пара. Погрешность для условий варочного цеха по экспериментальным данным составляет $\gamma_p = \pm 0,5\%$.

3. АЦП – аналого-цифровой преобразователь с линейно-изменяющимся напряжением. Погрешность основная относительная 02/01. Дополнительная погрешность из-за изменения температуры $T_{окр.ср} = (15 \div 40)^\circ\text{C}$ составит

$$\Delta\gamma_T = \frac{\pm 0,1\%}{\pm 10^\circ\text{C}}, \text{ дополнительная погрешность от изменения напряжения питания}$$

$\Delta\gamma_u = \pm 0,1\%$.

4. Контроллер типа Simatik S7-300 или Simatik S7-700. Погрешность представления температуры $\delta = \pm 0,05\%$.

Вычисления при $X = T_{\max}$

1. Как показано выше, для ТСПУ 014 полная погрешность ТЭС в реальных условиях эксплуатации на варочном котле составит

$$\Delta_p^1 = \pm 1,6^\circ\text{C},$$

или в приведенной форме

$$\gamma_p^1 = \frac{\pm \Delta_p \cdot 100\%}{T_w (= T_{\max})} = \frac{\pm 1,6^\circ\text{C} \cdot 100}{200^\circ\text{C}} = \pm 0,8\%.$$

2. Для ДЛС приведенная погрешность

$$\gamma_p^2 = \pm 0,5\%$$

3. Для АЦП основная относительная погрешность составит

$$\delta = \pm \left[c + d \left| \frac{X_k}{X} - 1 \right| \right]$$

или $\delta^3 = \pm \left[0,2 + 0,1 \left| \frac{200}{200} - 1 \right| \right] = \pm 0,2\%$, где $c = \pm 0,2\%$, $d = 0,1\%$, $X_k = T_{\max} = 200^\circ\text{C}$, и в

приведенной форме, так как $\delta = \frac{\pm \Delta^3 \cdot 100}{X}$ и $X = T_{\max}$ по условию расчета, то

$$\gamma^3 = \frac{\pm \Delta^3 \cdot 100}{T_{\max}} = \delta^3 = \pm 0,2\%$$

4. Контроллер имеет приведенную погрешность представления

$$\delta^4 = \gamma^4 = \pm 0,05.$$

Тогда результирующие погрешности ИК составят

$$\gamma_{\text{РИК}} \Big|_{\min} = \pm \left[(\delta^1)^2 + (\delta^2)^2 + (\delta^3)^2 + (\delta^4)^2 \right]^{1/2} = \pm \left[0,8^2 + 0,5^2 + 0,2^2 + 0,05^2 \right]^{1/2},$$

$$\gamma_{\text{РИК}} \Big|_{\max} = \pm (\delta + \delta + \delta + \delta) = \pm (0,8 + 0,5 + 0,2 + 0,05) = \pm 1,55\%$$

Следовательно, погрешность ИК лежит в интервале

$$\gamma_{\text{РИК}} \Big|_{\min} < \gamma_{\text{РИК}} < \gamma_{\text{РИК}} \Big|_{\max} = 1 < \gamma_{\text{РИК}} < \pm 1,55\%.$$

Это следует подтвердить при метрологической аттестации и поверке ИК при опытно-промышленной эксплуатации канала ИИС

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



УТВЕРЖДАЮ

И. о. проректора по учебно-методической работе _____ В. В. Зубов

Лабораторный практикум

Б1.В.09.06 СРЕДСТВА ИЗМЕРЕНИЙ

Специальность -

21.05.04 Горное дело

Специализация -

Мехатроника и робототехника промышленных комплексов

Одобрена на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой _____

(подпись)

Волков Е.Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического

(название факультета)

Председатель _____

(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА 1 «ИЗУЧЕНИЕ ДАТЧИКА ИЗБЫТОЧНОГО ДАВЛЕНИЯ»	3
ЛАБОРАТОРНАЯ РАБОТА 2 «ИЗУЧЕНИЕ ДАТЧИКА ДИФФЕРЕНЦИАЛЬНОГО ДАВЛЕНИЯ»	5
ЛАБОРАТОРНАЯ РАБОТА 3 «ИЗУЧЕНИЕ ТЕРМОПАРЫ»	8
ЛАБОРАТОРНАЯ РАБОТА 4 «ИЗУЧЕНИЕ ТЕРМОСОПРОТИВЛЕНИЯ»	12
ЛАБОРАТОРНАЯ РАБОТА №5 «ИЗУЧЕНИЕ РЕГУЛИРУЕМОГО БИМЕТАЛЛИЧЕСКОГО ТЕРМОСТАТА».....	23
ЛАБОРАТОРНАЯ РАБОТА №6 «ИЗУЧЕНИЕ ДАТЧИКОВ ЦВЕТА И СВЕТА»	26

ЛАБОРАТОРНАЯ РАБОТА 1 «ИЗУЧЕНИЕ ДАТЧИКА ИЗБЫТОЧНОГО ДАВЛЕНИЯ»

Цель работы – проведение исследований по работе прибора для измерения давления – датчика давления.

Теоретическая часть

Конструкция датчика избыточного давления представлена на рис.1. Преобразователь выполнен в цилиндрическом корпусе 3, в нижней части которого расположен штуцер 4, предназначенный для присоединения к линии измеряемого давления. В верхней части корпуса расположена «обойма» 9, которая крепится в корпусе с помощью специальных защелок, позволяющих ей вращаться вокруг своей оси (относительно корпуса 3). Для фиксации положения обоймы относительно корпуса служит крышка 10, которая навинчивается на наружную резьбу верхней части корпуса 3. На обойме установлена приборная часть 2 электрического соединителя типа DIN43650С. В кабельной части 1 соединителя производится подсоединение проводов внешних электрических цепей с помощью винтовых зажимов (клемм) без применения пайки. Во входном отверстии 5 приемной полости штуцера преобразователя предусмотрена резьба для установки гидравлического дросселя, предназначенного для предотвращения повреждения мембраны чувствительного элемента преобразователя в случае возникновения гидроудара.

В штуцере преобразователя размещен чувствительный элемент 6. В качестве чувствительного элемента применен тензопреобразователь, на котором размещена тензочувствительная полупроводниковая схема из четырех тензорезисторов, соединенных в мост Уитсона.

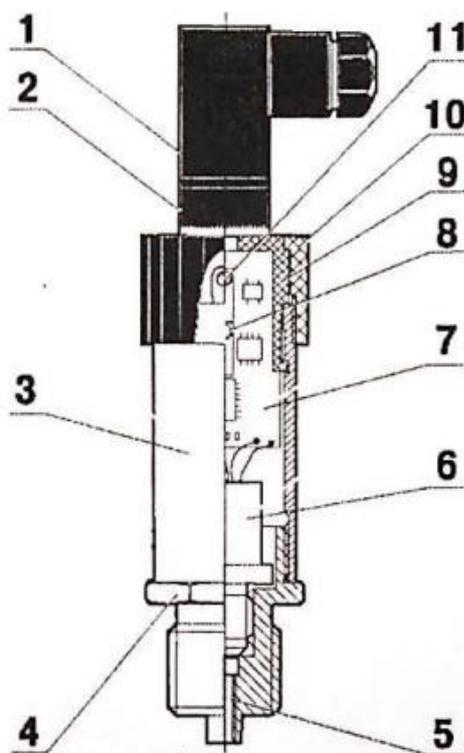


Рисунок 1

Под действием давления измеряемой среды мембрана чувствительного элемента прогибается. Тензорезисторы, деформируясь, изменяют свое сопротивление. В результате происходит разбалансировка моста пропорционально измеряемому давлению. Разбалансированный сигнал в виде

электрического сигнала преобразуется электронным блоком, расположенным в корпусе преобразователя, в выходной унифицированный сигнал постоянного тока 4...20 мА.

Порядок выполнения работы:

1. Изучить теоретические основы по материалам лекций или перечню рекомендованной преподавателем литературы.
2. Убедиться, что все выключатели модулей стенда находятся в положении «ВЫКЛ»
3. Выбрать модули стенда для выполнения текущего задания. Расставить их на лабораторной стойке так, чтобы было удобно проводить эксперимент. Подготовить соединительные провода (перемычки), входящие в комплект поставки стенда.
4. Подключить защитное заземление.
5. Подключить модули стенда к сети ~220В 50Гц.
6. Соединить модули стенда согласно принципиальной электрической схеме или схеме соединений.
7. Провести эксперимент.
8. Отключить модули от сети ~220В 50Гц.
9. Составить отчет по лабораторной работе.

Порядок выполнения эксперимента:

1. Подготовить стенд к работе
2. Включить компрессор.
3. Подключить пневматическую трубку от компрессора в быстросъемный разъем на тыльной стенке модуля набор датчиков давления а1.
4. С помощью регулятора давления установить величину давления 0,1 МПа на компрессоре.
5. Занести показания манометра и датчика давления в таблицу 1
6. С помощью регулятора давления увеличить значение давления на 0,1 МПа.
7. Занести показания манометра и датчика давления в таблицу
8. Повторить пункты 4-5 до достижения давления 0,4 МПа.
9. Выключить компрессор. Принять показания манометра за истинное значение давления. Вычислить абсолютную и относительную погрешность датчика давления.
10. Проанализировать результаты, сделать выводы по проделанной работе
11. Подготовить отчет и представить его преподавателю.

Таблица 1 Результаты исследования работы датчика давления

Номер измерения	Р манометра кПа	Р датчика давления кПа	ΔР датчика давления кПа
1			
2			
3			
4			
5			

ЛАБОРАТОРНАЯ РАБОТА 2 «ИЗУЧЕНИЕ ДАТЧИКА ДИФФЕРЕНЦИАЛЬНОГО ДАВЛЕНИЯ»

Цель работы: проведение исследований по работе прибора для измерения давления – датчика дифференциального давления.

Теоретическая часть

Датчики дифференциального (перепада) давления применяются для преобразования перепада давления (разности давлений) в унифицированный выходной сигнал напряжения, тока или индуктивности. Наиболее популярными являются датчики с унифицированным токовым выходом 4-20 мА.

Устройство датчика и принцип работы.

Основным элементом датчика дифференциального давления служит измерительная мембрана, изготовленная из упругого металла. К ней крепятся чувствительные тензорезисторы, которые способны измерять величину ее деформации. Мембрану датчика и рабочая среда изолированы друг от друга, через защитные эластичные перепонки к ней передается давление. Специальная жидкость заполняет полости, которые расположены между ними.

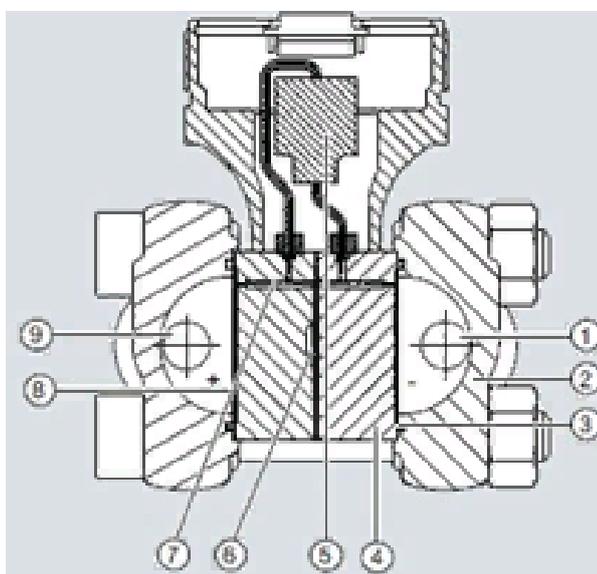


Рисунок 1 - Устройство датчика дифференциального давления: 1 - входное давление P ; 2 - фланцевые соединения подключениями к процессу; 3 - О-кольцо; 4 - корпус измерительной ячейки ; 5 - тонкий металлический проводник ; 6- перегрузочная мембрана ; 7 - наполнительная жидкость ; 8 - разделительная мембрана ; 9 - входное давление P+

Датчики, способные измерять дифференциальное давление, работают следующим образом: на защитные перепонки прибора воздействует давление рабочей среды, что в виде импульса передается на измерительную мембрану, которая оказывает влияние на тензисторы и влечет за собой изменение величины сопротивления. Электроника датчика фиксирует подобные модификации и на выходе преобразует их в аналоговый сигнал определенного значения.

Особенности приборов

Дифференциальные датчики от обычных преобразователей напора отличает ряд характерных отличительных черт. У дифференциальных датчиков 2 защитные мембраны, поэтому давление передается с двух сторон одновременно. В их конструкции предусмотрены 2 штуцера: минусовой и плюсовой. Когда измеряется

давление в рабочей среде, и показатели от напора на штуцер со знаком плюс превышают показатели минусового штуцера, то разница будет зафиксирована прибором с плюсовым значением, и наоборот.

Области применения приборов.

Сферы применения датчиков давления на сегодняшний день достаточно широка. Приборы данного типа используют в системах газоснабжения, вентиляции, водоподготовки, в нефтехимической, химической промышленности, на фармацевтическом производстве.

Измерение разницы напора в различных жидкостных средах необходимо для нормальной работы вентиляционных систем, а также устройств кондиционирования. Датчиками дифференциального давления пользуются для контроля загрязнённости систем фильтрации, управления демпферами, для регулирования и индикации воздушных потоков.

Принцип работы и строение датчика давления.

Главным элементом для такого датчика служит измерительная мембрана. Она изготавливается из упругого материала. Измеряется величина её деформации при помощи чувствительных тензорезисторов. Сама мембрана находится в изоляции от рабочих сред, давление на нее, оказывается, через защитные эластичные перепонки. Расположенные между мембраной и перепонками полости заполнены специальной жидкостью.

Давление оказывается из рабочей среды на перепонки, а они, в свою очередь, дают на измерительную мембрану. Мембрана передаёт импульс тензорезисторам, где меняется величина сопротивления. Эти слабые колебания улавливает встроенная электроника и преобразует их в аналоговый сигнал на выходе.

Особенности данных датчиков.

Отличием от прочих преобразователей напора является то, что дифференциальные датчики имеют две защитные мембраны и измеряют давление с двух сторон сразу. Также в них установлены плюсовой и минусовой штуцеры. Таким образом, прибор отражает разницу давления по обе стороны датчика.

Сферы применения приборов.

Данные датчики применяются при устройстве систем вентиляции, водоподготовки, газоснабжения.

Порядок выполнения работы:

1. Изучить теоретические основы по материалам лекций или перечню рекомендованной преподавателем литературы.
2. Убедиться, что все выключатели модулей стенда находятся в положении «ВЫКЛ»
3. Выбрать модули стенда для выполнения текущего задания. Расставить их на лабораторной стойке так, чтобы было удобно проводить эксперимент. Подготовить соединительные провода (перемычки), входящие в комплект поставки стенда.
4. Подключить защитное заземление.
5. Подключить модули стенда к сети ~220В 50Гц.
6. Соединить модули стенда согласно принципиальной электрической схеме или схеме соединений.
7. Провести эксперимент.
8. Отключить модули от сети ~220В 50Гц.

9. Составить отчет по лабораторной работе.

Порядок выполнения эксперимента:

1. подготовить стенд к работе
2. включить компрессор.
3. подключить пневматическую трубку от компрессора в быстросъемный разъем на тыльной стенке «модуля набор датчиков давления» А2.
4. При помощи регулятора давления увеличить значение давления на 0,1 МПа на компрессоре.
5. Занести показания манометра и датчика давления А2 в таблицу
6. Повторить пункты 4-5 до достижения давления 0,4 МПа.
7. Выключить компрессор.
8. Проанализировать результаты, сделать выводы по проделанной работе
9. Подготовить отчет и представить его преподавателю.

Номер измерения	Р манометра кПа	Р А2 датчика давления кПа
1		
2		
3		
4		
5		

ЛАБОРАТОРНАЯ РАБОТА 3 «ИЗУЧЕНИЕ ТЕРМОПАРЫ»

Цель работы: изучение термопар как устройства для контактного контроля над температурой объекта.

Теоретическая часть

Термопара – термоэлемент, представляющий собой пару проводников из различных материалов, соединенных на одном конце. На свободных концах в результате действия термоэлектрического эффекта наводится ЭДС.

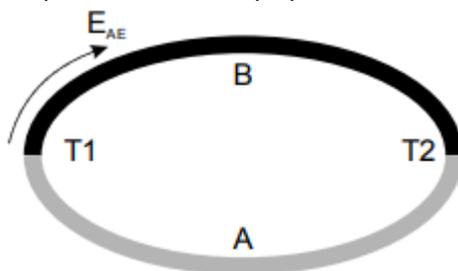


Рисунок 1

Термоэлектрический эффект, он же Эффект Зеебека – явление возникновения ЭДС в замкнутой электрической цепи (рис.1), состоящей из последовательно соединённых разнородных проводников (А и В), контакты между которыми находятся при различных температурах (Т1 и Т2).

Непосредственное практическое значение применительно к термопарам имеют следующие термоэлектрические законы.

Первый закон:

Возникающее в термопаре термо-ЭДС зависит от температуры спаев и не зависит от распределения температур по длине проводника (закон промежуточных температур Магнуса).

Иная его формулировка: термо-ЭДС термопары АВ с температурами спаев Т1 и Т2 равна алгебраической сумме двух термо-ЭДС этой же термопары с температурами спаев Т1 и Т3, Т3 и Т2:

$$E_{AA}(T_1 T_2) = E_{AA}(T_1 T_3) + E_{AA}(T_3 T_2) \quad (1)$$

Это правило позволяет не заботиться о термостатировании термопары по всей ее длине и используется для градуировки термопар и определения поправок на температуру холодного спая.

Второй закон:

Если цепь состоит из трех проводников АВС, то термо-эдс термопары АВ со спаями при температурах Т1 и Т2 равняется алгебраической сумме термо-эдс термопар АС и СВ, спаи которых находятся при тех же температурах.

$$E_{AA}(T_1 T_2) = E_{AA}(T_1 T_2) + E_{CC}(T_1 T_2) \quad (2)$$

Этот закон ("промежуточных металлов") позволяет использовать удлинительные концы к термопарам и присоединять термопару к приборам с другими проводниками.

Нетрудно сообразить, что присоединенный в термостате к обоим проводам (А и В) термопары медный провод, который ведет к измерительному прибору (провод С в приведенной формуле) не меняет величины возникшей

электродвижущей силы термопары АВ, поскольку эффекты от спаев СА и СВ, находящихся при одной температуре компенсируют друг друга.

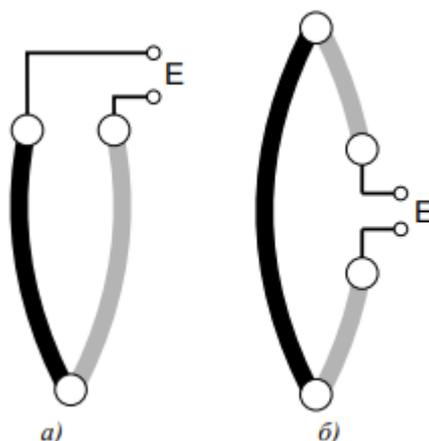


Рисунок 2

Таким образом, термопара может образовывать устройство или его часть, использующее термоэлектрический эффект для измерения температуры. В сочетании с электроизмерительным прибором термопара образует термоэлектрический термометр. Измерительный прибор или электронную измерительную систему подключают либо к концам термоэлектродов (рис. 2,а), либо в разрыв одного из них (рис. 2,б). В местах подключения проводников термопары к измерительной системе возникают дополнительные термо-ЭДС, которые взаимно компенсируются при равенстве температур вспомогательных спаев.

Так как термо-ЭДС термопары зависит от разности температур холодного и горячего концов,

для успешного преобразования ЭДС в температуру необходимо так же контролировать температуру холодного спая. На практике при измерении температур широко используется техника «компенсации холодного спая»: температура холодного спая измеряется другим датчиком температуры, а затем величина термо-ЭДС холодного спая программно или аппаратно вычитается из сигнала термопары.



Рисунок 3

Места подключения термопары к измерительной системе должны иметь одинаковую температуру, то есть находиться в изотермальной зоне. Кроме того, в схеме с компенсацией холодного спая в этой же зоне должен находиться и датчик температуры холодного спая. При конструировании измерительной системы эти требования должны учитываться.

При выборе термопары для производства замеров температуры в некотором диапазоне следует выбирать ту термопару, коэффициент линейности которой изменяется менее других в рамках этого диапазона. Для достижения высокой

точности измерений термопарного термометра во всем диапазоне рабочих температур необходима его калибровка.

Среди основных можно выделить следующие типы термопар:

ТХА (хромель-алюмелевая термопара) – предназначена для работы в окислительных и инертных

средах. Обладает более близкой к прямой характеристикой.

ТХК (хромель-копелевая термопара) – предназначена для работы в окислительных и инертных средах. Обладает наибольшей чувствительностью, высокой термоэлектрической стабильностью при температурах до 600°C. Недостаток: высокая чувствительность к деформациям.

ТПП (платинородий-платиновая термопара) – предназначена для длительной эксплуатации в

окислительных средах. Обладает хорошей устойчивостью к газовой коррозии (особенно на воздухе при высоких температурах); высокой надежностью при работе в вакууме, но менее стабильны в нейтральных средах. Недостаток: высокая чувствительность термоэлектродов к любым загрязнениям, появившимся при изготовлении, монтаже или эксплуатации термопар.

ТВР (вольфрам-рениевая термопара) – может использоваться как стандартные для установления

номинальных статических характеристик термопар методом сравнения. Обладает возможностью длительного применения при температурах до 2200°C в неокислительных средах; устойчивостью в аргоне, гелии, сухом водороде и азоте. Недостаток: плохая воспроизводимость термо-ЭДС, вынуждающая группировать термоэлектродные пары по группам с номинальными статическими характеристиками.

ТНН (нихрасил-нисиловая термопара) – предназначена в качестве универсального средства измерения температур в диапазоне температур 0 - 1230°C. Обладает высокой стабильностью термо-ЭДС (по сравнению с ТХА, ТПП, ТПР), высокой радиационной стойкостью, высокой стойкостью к окислению электродов.

В зависимости от конструкции и назначения различают термопары погружаемые и поверхностные; с обыкновенной, взрывобезопасной, влагонепроницаемой или иной оболочкой (герметичной или негерметичной), а также без оболочки; обыкновенные, вибротряскоустойчивые и ударопрочные; стационарные и переносные и т.д.

Порядок выполнения работы:

1. Изучить теоретические основы по материалам лекций или перечню рекомендованной преподавателем литературы.

2. Убедиться, что все выключатели модулей стенда находятся в положении «ВЫКЛ»

3. Выбрать модули стенда для выполнения текущего задания. Расставить их на лабораторной стойке так, чтобы было удобно проводить эксперимент. Подготовить соединительные провода (перемычки), входящие в комплект поставки стенда.

4. Подключить защитное заземление.

5. Подключить модули стенда к сети ~220В 50Гц.

6. Соединить модули стенда согласно принципиальной электрической схеме или схеме соединений.

7. Провести эксперимент.

8. Отключить модули от сети ~220В 50Гц.

9. Составить отчет по лабораторной работе.

Порядок проведения эксперимента

Все исследования датчиков температуры проводятся в диапазоне: от температуры окружающей среды до 80°C°.

Ознакомиться с электрической схемой соединения датчика с измерительным прибором для снятия статической характеристики. Статическая характеристика датчика - функциональная зависимость между изменениями входной X и выходной Y величинами.

Собрать схему проведения эксперимента по исследованию датчика «ТЕРМОПАРА».

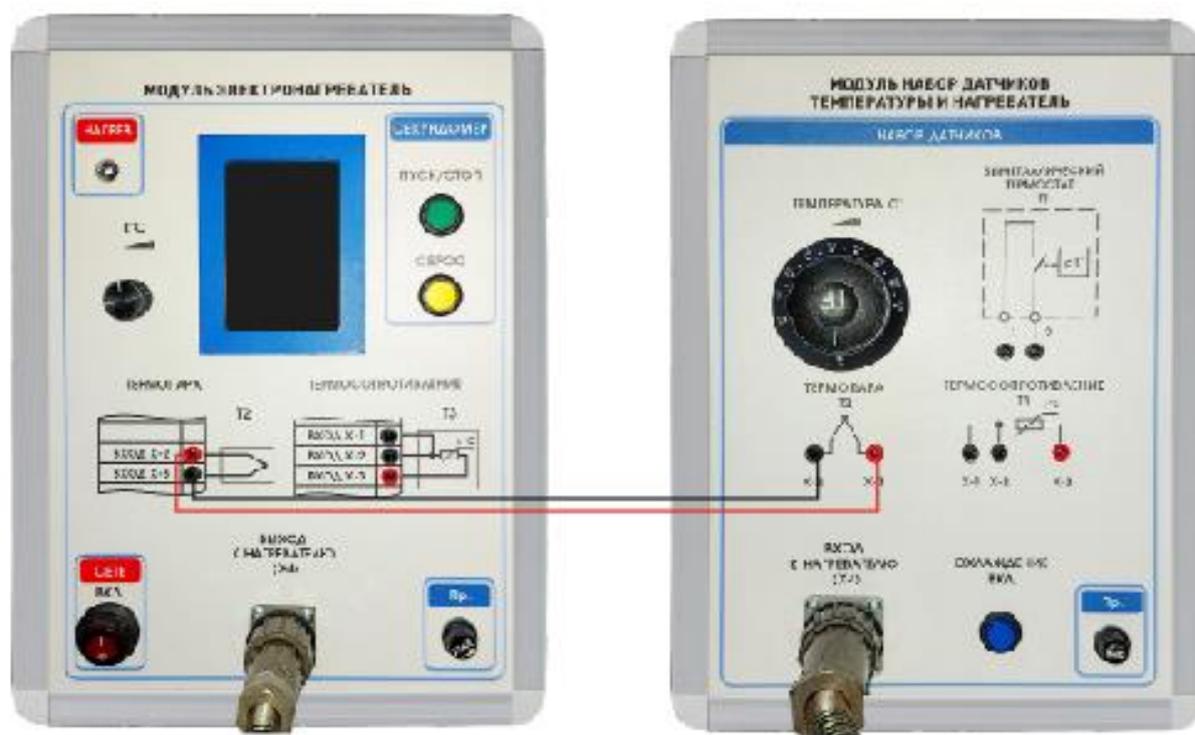


Рисунок 4

Представить схему для проверки преподавателю.

Включить питание модулей.

Включить нагревательный ТЭН.

Включить питание модулей. Произвести измерение напряжения на выходных клеммах термопары (0,2 – 2,0 мВ), изменяя температуру нагрева датчика от 25°C до 75°C.

По мере увеличения температуры записывать показания с дисплея прибора и показания величины показания напряжения с прибора через 1...2 °С.

Данные занести в таблицу и построить график зависимости Э.Д.С термопары от температуры.

Таблица.

	1	2	3	4	5	...	N
Время, сек							
Температура, °С							
Напряжение, мВ							

ЛАБОРАТОРНАЯ РАБОТА 4 «ИЗУЧЕНИЕ ТЕРМОСОПРОТИВЛЕНИЯ»

Цель работы: изучение динамических характеристик терморезистивного преобразователя.

Теоретическая часть

Термосопротивления представляют собой измерительный преобразователь с чувствительным элементом (проводник или полупроводник), который под воздействием температуры изменяет активное сопротивление электрическому току. Изменение электросопротивления данного материала при изменении температуры характеризуется температурным коэффициентом сопротивления α ($1/^\circ\text{C}$), который определяется по формуле:

$$\alpha = \frac{R_t - R_0}{R_0 t} \quad (1)$$

где t – температура материала, $^\circ\text{C}$; R_0 и R_t – электросопротивление при 0°C и при температуре t , Ом.

Проводниковые ТС

Материалы, которые используют для изготовления проводниковых ТС должны быть устойчивыми к нагреванию, что обуславливает однозначность зависимости сопротивления от температуры и стойкость проводника против коррозии и, следовательно, достаточную точность измерений; иметь большой α обеспечивающий высокую чувствительность прибора; значительное удельное сопротивление проводника позволяющее изготовить термометры малых размеров; линейность характеристики.

Наиболее полно указанным требованиям отвечают платина (интервал температур – $260 \div +750^\circ\text{C}$) и медь (интервал – $60 \div +180^\circ\text{C}$).

Электрическое сопротивление проволоки платиновых термометров сопротивления в интервале температур от 0 до $+650^\circ\text{C}$ определяется по формуле:

$$R_t = R_0 (1 + At + Bt^2) \quad (2)$$

где R_t – сопротивление термометра при температуре $t^\circ\text{C}$; R_0 – сопротивление термометра при температуре 0°C ; A и B – постоянные коэффициенты.

Для платины марки «Экстра»

$$R_{100} / R_0 = 1,391;$$

$$A = 3,96847 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1};$$

$$B = 5,847 \cdot 10^{-7} \text{ } ^\circ\text{C}^{-2}.$$

Сопротивление платины в интервале температур от -200 до 0°C изменяется по закону:

$$R_t = R_0 [1 + At + Bt^2 + Ct^3(t - 100)] \quad (3)$$

где $C = -4,22 \cdot 10^{-12} \text{ } ^\circ\text{C}^{-1}$.

У платины, применяемой для изготовления технических термометров $\alpha = 3,9 \cdot 10^{-3} \text{ } 1/^\circ\text{C}$.

Из уравнений 1 и 2 видно, что характеристики платиновых термометров нелинейны однако отклонение от линейной характеристики не превышает 5% в интервале от 0 до 500°C и 19% в интервале температур от –200 до 0°C.

Электрическое сопротивление проволоки медных термометров сопротивления в интервале температур от –50 до +180°C определяется по формуле:

$$R_t = R_0(1 + \alpha t) \quad (4)$$

где α – температурный коэффициент сопротивления меди ($\alpha=4,26 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1}$).

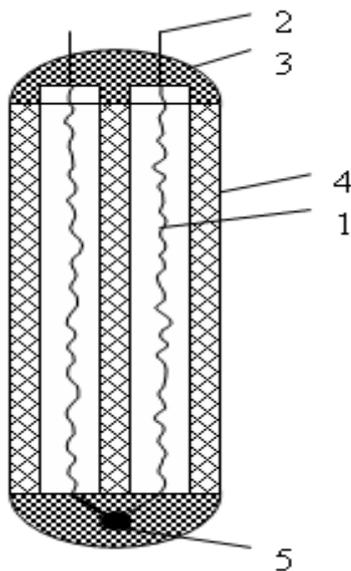


Рисунок 1 – Чувствительный элемент платинового ТС

К преимуществам меди следует отнести низкую стоимость, легкость получения ее в чистом виде, сравнительно высокий $\alpha=4,26 \cdot 10^{-3} \text{ } 1/^\circ\text{C}$ и линейную зависимость сопротивления от температуры.

К недостаткам меди относятся малое удельное сопротивление и легкая окисляемость при температуре выше 100°C.

Кроме того, для изготовления ТС используется никель и железо.

На рисунке показан чувствительный элемент платинового термометра. Он состоит из двух соединенных последовательно платиновых спиралей 1, расположенных в каналах керамического каркаса 4. К двум верхним концам этих спиралей припаяны платиновые или иридиево-родиевые (60% родия) выводы 2, к которым приварены выводные проводники изолированные керамическими бусами. Для крепления платиновых спиралей и выводов в керамическом каркасе используют глазурь (или термоцемент) 3 на основе окисей алюминия и кремния. Пространство между платиновыми спиралями и стенками каналов каркаса заполнено порошком окиси алюминия, который служит изолятором и улучшает тепловой контакт между спиралями и каркасом.

Для подгонки сопротивления при 0°C изменяют длину нижних концов платиновых спиралей с последующей пайкой 5.

Для защиты от воздействия химически агрессивных сред и механических повреждений чувствительные элементы помещают в кожух, выполненный из алюминия или нержавеющей стали. Все ТС выпускаются взаимозаменяемыми.

Термометры сопротивления ТСП выпускают с сопротивлением $R_0=1$ Ом (градуировка гр 1П), $R_0=10$ Ом (градуировка гр 10П) и $R_0=50$ Ом (градуировка гр 50П).

Термометры сопротивления ТСМ выпускают с сопротивлением $R_0=10$ Ом (градуировка гр 10М), $R_0=50$ Ом (градуировка гр 50М) и $R_0=100$ Ом (градуировка гр 100М).

Полупроводниковые ТС

Полупроводниковые термометры сопротивления, которые называются термисторами или терморезисторами, применяются для измерения температуры в интервале от -90 до $+180$ °С. Существенным их преимуществом является большой температурный коэффициент электрического сопротивления. Вследствие большого удельного электрического сопротивления полупроводников из них можно изготовить термометры малых размеров с большим начальным сопротивлением, что позволяет не учитывать сопротивление соединительных проводов и других элементов электрической измерительной схемы термометра.

Чувствительные элементы термисторов, изготавливаемые из окислов меди, марганца, магния, никеля, кобальта и других металлов, имеют форму небольших цилиндров, дисков или шариков.

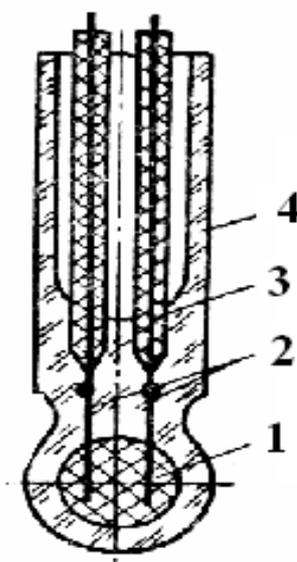


Рисунок 2 - Бусинковый полупроводниковый терморезистор

Сопротивление термисторов уменьшается с повышением температуры. Для большинства полупроводниковых сопротивлений зависимость электрического сопротивления от температуры имеет нелинейный характер и может быть в общем виде описана зависимостью:

$$R_T = A \cdot e^{B/T} \quad (4)$$

где R_T – электрическое сопротивление термистора при абсолютной температуре T , К; A и B – константы термистора.

На рисунке 2 показан бусинковый терморезистор, состоящий из полупроводникового элемента 1, защищенного стеклянной оболочкой 4. В шарик 1 вмонтированы электроды 2 из платиновой проволоки, соединенные с никелевыми выводами 3.

Основным препятствием, ограничивающим широкое внедрение полупроводниковых терморезисторов в промышленность, является плохая воспроизводимость их параметров, что исключает их взаимозаменяемость, а также сравнительно невысокая максимальная рабочая температура (от -60 до 180°C).

В качестве устройств, измеряющих электрическое сопротивление термометров в функции температуры, применяют мосты и логометры. Мосты бывают уравновешенные и неуравновешенные, автоматические и неавтоматические.

Принципиальная схема уравновешенного моста показана на рис. Здесь R_1 и R_3 – постоянные сопротивления, R_2 – переменное сопротивление (реохорд), с помощью которого мост приводят в равновесное состояние, R_T – сопротивление термометра, НП – нуль-прибор.

При равновесии мостовой схемы ток I_0 в измерительной диагонали равен нулю, а падения напряжений на сопротивлениях R_1 и R_3 должны быть равны между собой, т.е.

$$I_1 R_1 = I_3 R_3 \quad (5)$$

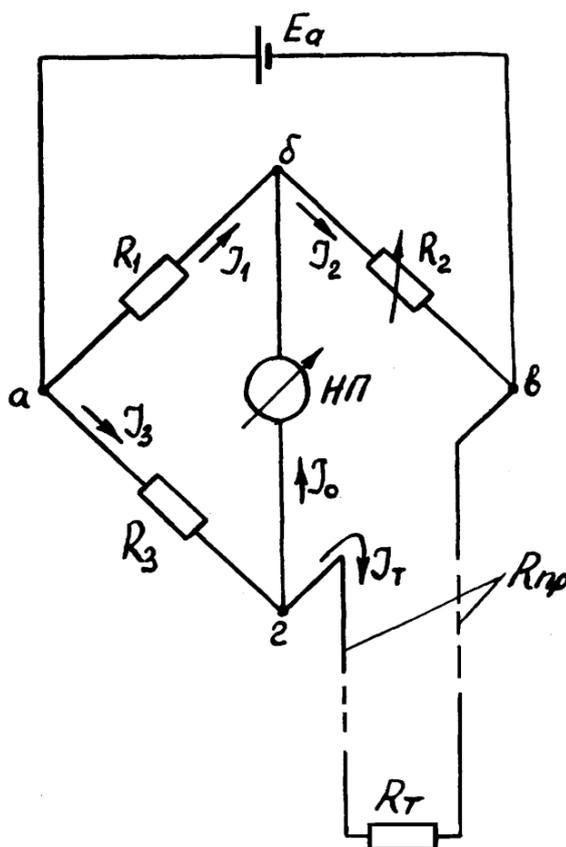


Рис. 3 – Принципиальная схема уравновешенного моста

Кроме того, падения напряжений на плечах «бв» и «вг» также должны быть равными между собой, т.е.

$$I_2 R_2 = I_T (R_T + R_{тр}) \quad (6)$$

где $R_{\text{пр}}$ – сопротивление соединительных проводов.

Разделив (6) на (7) и принимая во внимание, что при равновесии моста $I_1 = I_2$; $I_3 = I_T$, получим:

$$\frac{R_1}{R_2} = \frac{R_3}{R_T + R_{\text{пр}}} \quad (7)$$

или

$$R_2 \cdot R_3 = R_1 (R_T + R_{\text{пр}}) \quad (8)$$

Следовательно, условием равновесия моста является равенство произведений сопротивлений противоположных плеч.

Из равенства (9) сопротивление термометра R_T будет равно:

$$R_T = \frac{R_3}{R_1} R_2 - R_{\text{пр}} \quad (9)$$

Каждому значению R_T при равновесном состоянии мостовой схемы соответствует определенное значение R_2 . Шкалу реохорда R_2 градуируют в °С, что позволяет по положению движка определить температуру нагрева термометра.

Для исключения влияния на результаты измерений колебаний сопротивления соединительных проводов применяют трехпроводную схему включения термометра (рисунок).

В этом случае для симметричного моста (когда $R_1 = R_3$) при условии равенства сопротивлений соединительных проводов равновесие моста будет иметь место при $R_T = R_2$, т.е. изменение сопротивления проводов не будет влиять на результаты измерений.

Уравновешенные мосты имеют высокий класс точности, допускают колебания напряжения источника в пределах $\pm 20\%$, позволяют автоматизировать процесс измерения.

В неуравновешенном измерительном мосту тремя плечами являются постоянные сопротивления, а термометр сопротивления включается в четвертое плечо. К одной из диагоналей подключается источник питания, а в другую пиromетрический милливольтметр (ПМ), показания которого зависят от сопротивления термометра. Сопротивление R_k служит для контроля напряжения питания схемы.

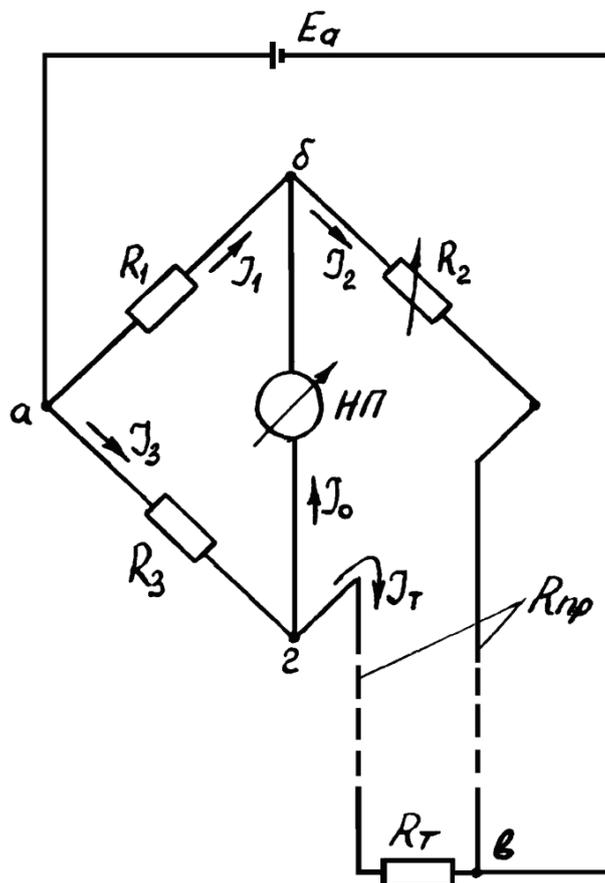


Рисунок 4 – Трехпроводная схема включения термометра сопротивления в уравновешенный мост

Перед началом измерений переключатель П ставят в положение 1. В этом случае четвертым плечом моста будет сопротивление R_k . Если питающее напряжение U_{ab} имеет номинальное значение, то стрелка милливольтметра устанавливается против контрольной отметки на шкале. Если этого нет, то, изменяя положение движка реостата R , выводят стрелку милливольтметра на контрольную отметку. При измерении переключатель ставят в положение 2. В этом случае четвертым плечом моста будет сумма сопротивлений $(R_T + R_{пр})$, ток через рамку милливольтметра будет функцией этой суммы сопротивлений. Отсчет результата измерения производят по заранее проградуированной шкале милливольтметра. Такие мосты применяют крайне редко, так как результаты измерений зависят от колебаний сопротивления соединительных проводов и питающего мост напряжения. Рекомендуется применять неуравновешенные мосты в тех случаях, когда сопротивление датчика во много раз превышает сопротивление соединительных проводов.

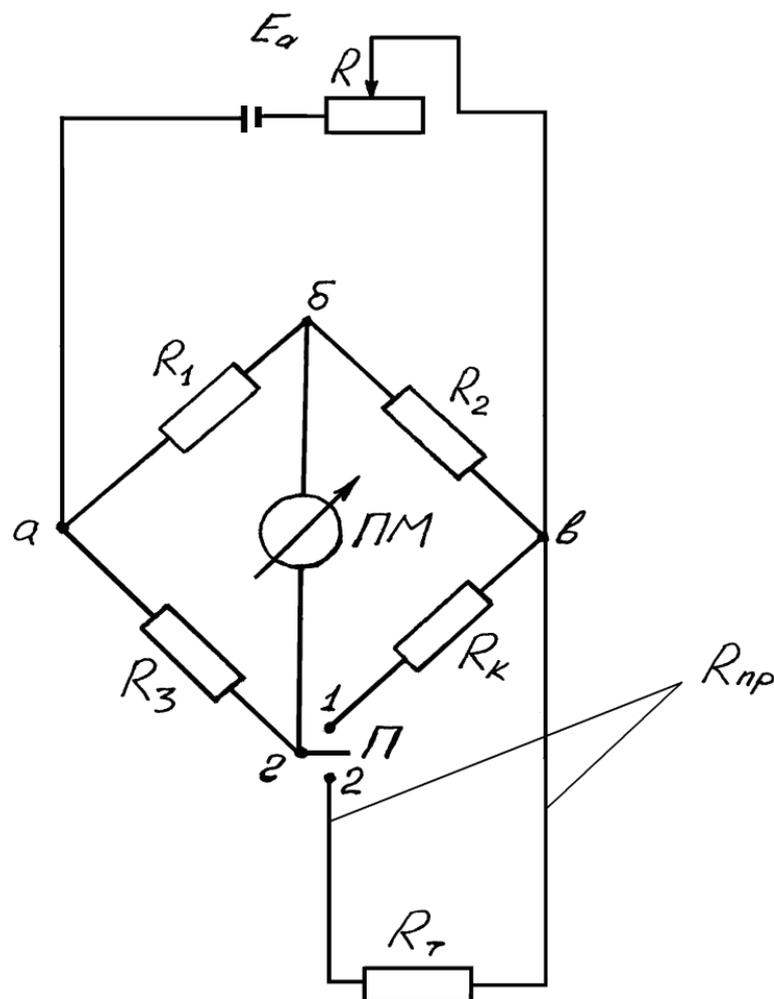


Рисунок 5 – Принципиальная схема неуравновешенного моста: E_a – источник питания; R – реостат для установки номинальной величины напряжения питания схемы; R_1, R_2, R_3 – постоянные сопротивления; Π – переключатель; R_k – контрольное сопротивление; R_T – сопротивление термометра; $R_{пр}$ – сопротивление проводов;

Автоматические уравновешенные мосты широко используются для измерения и регистрации температуры в комплекте с ТС, а также других параметров, изменение которых может быть преобразовано в изменение активного сопротивления. Их характеризуют высокая точность и возможность использования в системах автоматического регулирования. Они выпускаются различных модификаций: одно- и многоточечные, с дисковой и ленточной диаграммой, с сигнальными устройствами и др.

На рисунке приведена принципиальная схема автоматического уравновешенного моста, работающего на переменном токе, который также как ручной равновесный мост, реализует нулевой метод измерения сопротивления.

Напряжение разбаланса на вершинах моста a и b подается на вход электронного усилителя ЭУ. В нем оно усиливается до величины, достаточной для приведения в действие реверсивного электродвигателя РД. Ротор двигателя вращается в ту или другую сторону в зависимости от знака напряжения разбаланса, через систему передач перемещает движок компенсирующего переменного резистора (реохорда) R_p , уравновешивая измерительную схему моста, а также перемещает показывающую стрелку. Если мост находится в равновесии, то ротор РД не вращается, т.к. напряжения на вход электронного усилителя не подается.

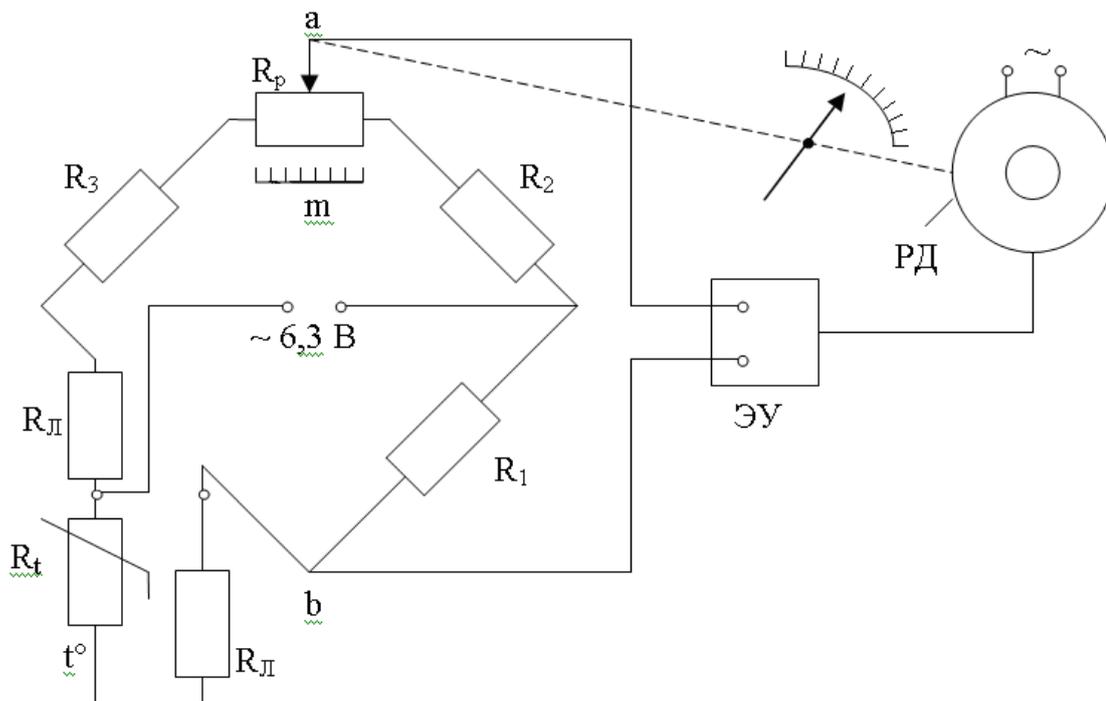


Рисунок 6 - Принципиальная схема автоматического уравновешенного моста

Кроме мостов, в качестве измерительных приборов для термометров сопротивления иногда применяют логометры, представляющие собой разновидность электроизмерительных приборов магнитоэлектрической системы (рисунок).

Постоянный магнит снабжен полюсными наконечниками N и S с цилиндрическими выточками. Центры выточек полюсных наконечников смещены относительно центра сердечника. Между полюсными наконечниками расположен цилиндрический сердечник из стали, вокруг которого вращается подвижная система из двух рамок R_1 и R_2 . К рамкам прикреплена стрелка, перемещающаяся вдоль шкалы. Воздушный зазор между полюсными наконечниками и сердечником неравномерный, поэтому магнитная индукция меняется (наибольшее значение в середине полюсных наконечников, наименьшее - у края), являясь функцией угла поворота от среднего положения.

К рамкам подводится ток от общего источника питания (G). В рамку R_1 ток поступает через термометр сопротивления R_t , в рамку R_2 – через резистор R постоянного сопротивления. Направление сил тока I_1 и I_2 таково, что вращающие моменты рамок оказываются направленными навстречу один другому.

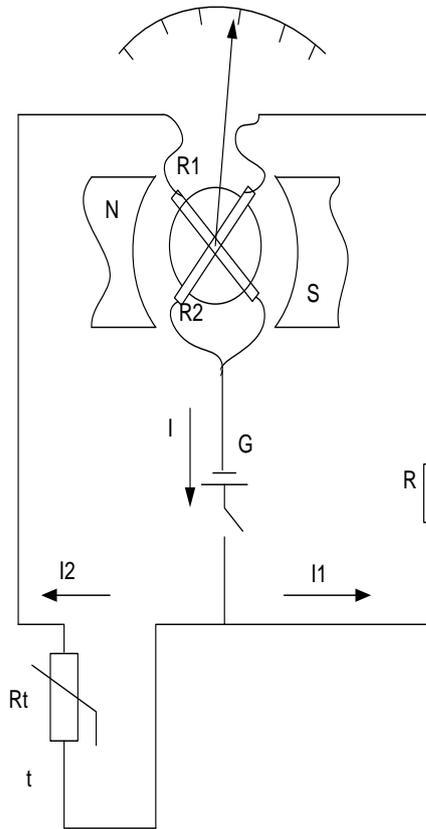


Рис. 7 - Принципиальная схема логометра

Вращающие моменты рамок:

$$M_1 = C_1 \cdot B_1 \cdot I, \quad (10)$$

$$M_2 = C_2 \cdot B_2 \cdot I_2, \quad (11)$$

где C_1 и C_2 – постоянные, геометрических размеров и числа витков рамок; B_1 и B_2 – индукции в зоне расположения рамок.

Если сопротивление рамок одинаковое $R_1 = R_2$ и $R = R_t$, то $I_1 = I_2$, т.е. вращающие моменты рамок $M_1 = M_2$. При этом подвижная система находится в среднем положении. При изменении сопротивления термометра вследствие нагрева (или охлаждения) через одну из рамок потечет ток большей силы, равенство моментов нарушится, и подвижная система начнет поворачиваться в сторону действия большего момента. При вращении подвижной системы рамка, по которой течет ток большей силы, попадает в зазор с меньшей магнитной индукцией, вследствие чего действующий на нее момент уменьшается. При этом другая рамка входит в зазор с большей магнитной индукцией, и ее момент увеличивается. Вращение рамок продолжается до тех пор, пока их вращающие моменты станут снова равны. В момент равновесия

$$C_1 \cdot B_1 \cdot I_1 = C_2 \cdot B_2 \cdot I_2 \quad (12)$$

или

$$\frac{I_1}{I_2} = \frac{C_2 \cdot B_2}{C_1 \cdot B_1} = C \cdot \frac{B_2}{B_1}. \quad (13)$$

Подставив в это уравнение значения I_1 и I_2 , выраженные через напряжение источника питания U и сопротивление цепей, получим:

$$\frac{\frac{U}{R+R_t}}{\frac{U}{R_t+R_2}} = \frac{R_t+R_2}{R+R_1} = C \frac{B_2}{B_1}. \quad (14)$$

Так как $B = f(\varphi)$, то и отношение $B_2/B_1 = f(\varphi)$; тогда

$$\frac{R_t+R_2}{R+R_1} = f(\varphi) \quad (15)$$

или

$$\varphi = f\left(\frac{R_t+R_2}{R+R_1}\right). \quad (16)$$

В этом выражении R , R_1 и R_2 постоянные, поэтому угол поворота подвижной системы зависит только от сопротивления термометра: $\varphi = f(R_t)$.

Показания прибора не зависят от колебания напряжения источника питания только в определенных границах. Так, при колебаниях напряжения питания $\pm 20\%$ возникает угловая погрешность $\Delta\varphi \approx 0,5 - 1\%$.

К рамкам логометра ток подводится тремя тонкими спиральными волосками, служащими одновременно для возвращения стрелки прибора к началу шкалы при прекращении питания прибора током.

Порядок выполнения работы:

1. Изучить теоретические основы по материалам лекций или перечню рекомендованной преподавателем литературы.
2. Убедиться, что все выключатели модулей стенда находятся в положении «ВЫКЛ»
3. Выбрать модули стенда для выполнения текущего задания. Расставить их на лабораторной стойке так, чтобы было удобно проводить эксперимент. Подготовить соединительные провода (перемычки), входящие в комплект поставки стенда.
4. Подключить защитное заземление.
5. Подключить модули стенда к сети $\sim 220\text{В}$ 50Гц.
6. Соединить модули стенда согласно принципиальной электрической схеме или схеме соединений.
7. Провести эксперимент.
8. Отключить модули от сети $\sim 220\text{В}$ 50Гц.
9. Составить отчет по лабораторной работе.

Порядок выполнения эксперимента:

Ознакомиться с электрической схемой соединения датчика с измерительным прибором для снятия статической характеристики. Статическая характеристика датчика - функциональная зависимость между изменениями входной X и выходной Y величинами.

Собрать схему пр8

Представить схему для проверки преподавателю.

Включить питание модулей. Произвести измерение сопротивления на выходных клеммах датчика (2 – 10 Ом), изменяя температуру нагрева датчика от 25°C до 75°C

По мере увеличения температуры записывать показания с дисплея прибора и показания величины показания напряжения с прибора (мультиметра) через 1...2 °С.

Данные занести в таблицу и построить график зависимости сопротивления от температуры.

	1	2	3	4	5	...	N
Время, сек							
Температура, °С							
Сопротивление мОм							

ЛАБОРАТОРНАЯ РАБОТА №5 «ИЗУЧЕНИЕ РЕГУЛИРУЕМОГО БИМЕТАЛЛИЧЕСКОГО ТЕРМОСТАТА»

Цель работы – изучение биметаллического регулируемого термостата.

Теоретическая часть:

Впервые биметаллические пластины были применены в XVIII веке английским часовщиком Джоном Харрисоном, который использовал их для регулирования точности хода морских хронометров. Современный биметаллический терморегулятор – это наиболее практичное и дешевое решение проблемы контроля рабочего диапазона температур.

Принцип работы биметаллического термостата.

При нагреве металлы расширяются. Каждый имеет свой коэффициент линейного расширения, из-за чего происходит искажение размеров и формы изделий из них. Это свойство определяет механизм работы биметаллических термостатов, который имеет такую последовательность:

Происходит нагрев пластины, один из слоев которой удлиняется быстрее.

В зависимости от видов металлов происходит постепенный или мгновенный изгиб пластины. Второй вариант изменения ее формы также называется механической бифуркацией.

Пластина приводит в действие исполнительный механизм термостата, который разрывает цепь.

Пластина остывает и возвращается в первоначальное состояние. Контакты устройства срабатывают автоматически или принудительно через кнопку, включая при этом цепь.

По своей сути регулируемый биметаллический термостат является тепловым реле, препятствующим перегреву оборудования. Такие устройства обычно рассчитаны на относительно небольшие температуры (до 180°C). Они также имеют нерегулируемое исполнение.

Преимущества использования биметаллических терморегуляторов.

Тепловые реле имеются в счетчиках, автоматических выключателях, электромоторах, бытовой технике и различном промышленном оборудовании. Биметаллический терморегулятор имеет несомненные достоинства: предельную простоту и надежность – устройство выдерживает до 100 тыс. циклов срабатывания; высокую точность срабатывания – $\pm 1^\circ\text{C}$; низкую цену – биметаллическое реле в разы, а иногда и на порядок дешевле термостатов капиллярного типа; компактность, энергонезависимость и прочее.

Недостатком является инерционность – устройство может срабатывать с некоторым запозданием, но эта проблема решается путем настройки реле на более слабый нагрев.

Порядок выполнения работы:

1. Изучить теоретические основы по материалам лекций или перечню рекомендованной преподавателем литературы.

2. Убедиться, что все выключатели модулей стенда находятся в положении «ВЫКЛ»

3. Выбрать модули стенда для выполнения текущего задания. Расставить их на лабораторной стойке так, чтобы было удобно проводить эксперимент. Подготовить соединительные провода (перемычки), входящие в комплект поставки стенда.

4. Подключить защитное заземление.

5. Подключить модули стенда к сети ~220В 50Гц.

6. Соединить модули стенда согласно принципиальной электрической схеме или схеме соединений.
7. Провести эксперимент.
8. Отключить модули от сети ~220В 50Гц.
9. Составить отчет по лабораторной работе.

Порядок выполнения эксперимента:

1. Собрать схему проведения эксперимента по исследованию датчика «биметаллический термостат».

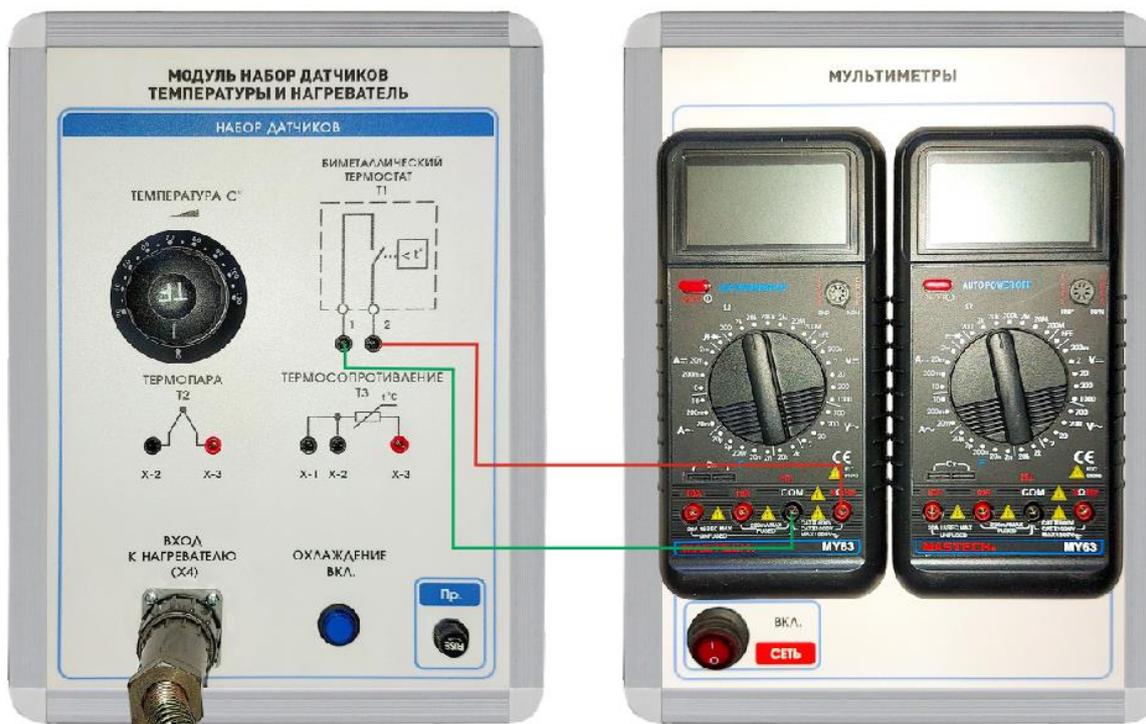


Рис. 9

2. Представить схему для проверки преподавателю.
3. Включить питание модулей. На модуле «МУЛЬТИМЕТРЫ» установить режим



(прозвонки)

4. выставить температуру на биметаллическом термостате 30 С°
5. Выставить уставку по температуре 75 С° на блоке «модуль электронагреватель»
6. Засечь на секундомере за какое время нагреется электронагреватель и замкнутся концы биметаллического термостата.
7. Повторить пункты 4-6 для 40,50,60 ,70 С°
8. Записать показания в таблицу
9. Построить график зависимости температуры срабатывания термостата от времени

Таблица

Температура уставки теромостата, °C	30C°	40C°	50C°	60C°	70C°
Температура уставки нагревателя, °C	75C°	75C°	75C°	75C°	75C°
Время, сек					

ЛАБОРАТОРНАЯ РАБОТА №6 «ИЗУЧЕНИЕ ДАТЧИКОВ ЦВЕТА И СВЕТА»

1.1. Цель работы:

1. Изучение датчика цвета.
2. Изучение фотодиода и влияния силы света на его характеристики.
3. Изучение фоторезистора и влияния силы света на его характеристики.
4. Изучение фототранзистора и влияния силы света на его характеристики.

1.2. Теоретические сведения

1. Датчик цвета TCS3200 можно использовать для распознавания цветов окружающих предметов в роботизированных системах. Датчик состоит из четырех светочувствительных сенсоров с оптическими фильтрами – красным, зеленым, голубым и бесцветным. Это позволяет выделить из света, отражающегося от исследуемого материала, четыре цветовых компонента.

Также на датчике располагаются четыре белых светоизлучающих диода, которые нужны для подсветки исследуемой поверхности.



Рисунок 1.2.1

Основа модуля определения цвета RKP-TSC3200 это микросхема TCS3200 (TAOS TCS3200 RGB sensor chip) представляет собой матрицу из фотодиодов 8 x 8, всего 64 фотодиода в одном кристалле. Из них 16 фотодиодов имеют синий светофильтр, 16 фотодиодов имеют зеленый светофильтр, 16 фотодиодов имеют красный светофильтр и 16 фотодиодов без светофильтра.

Такая внутренняя компоновка микросхемы TCS3200 позволяет датчику измерять интенсивность цвета в четырех цветовых диапазонах: синем, зеленом, красном и белом.

Питание датчика цвета необходимо осуществлять от источника питания напряжением +5 В. Управлять работой датчика можно сигналами S0, S1, S2, S3 и S4.

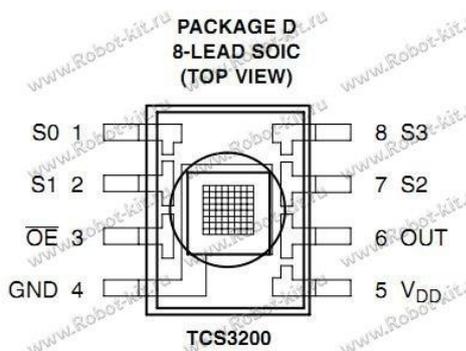


Рисунок 1.2.2

Датчик выдает информацию о цвете в виде импульсов, частота которых пропорциональна интенсивности света поступающего на соответствующий сенсор. Для удобства исследований цифровой выходной сигнал преобразован в аналоговый при помощи ЦАП.

2. Фотодиод - прибор, основанный на явлении фотовольтаического эффекта в полупроводниковом контактном переходе и предназначенный как для работы с приложением внешнего напряжения, так и без него.

В настоящее время при создании фотодиодов чаще других применяются p-n переходы, поэтому рассмотрим работу таких фотодиодов.

Фотодиод представляет собой пластинку полупроводникового материала, внутри которого имеются области примесной электронной (n – область) и дырочной (p – область) проводимостей. Границу между этими областями называют контактным p-n переходом (рисунок 1.2.3). Электронная и дырочная области снабжены невыпрямляющими контактами с присоединенными к ним выводами, с помощью которых осуществляется связь с внешней цепью. С целью предохранения чувствительного слоя фотодиода от воздействия внешней среды он покрывается лаком или монтируется в герметичном корпусе, изготовленном или из металла (со стеклянным входным окном) или из пластмасс.

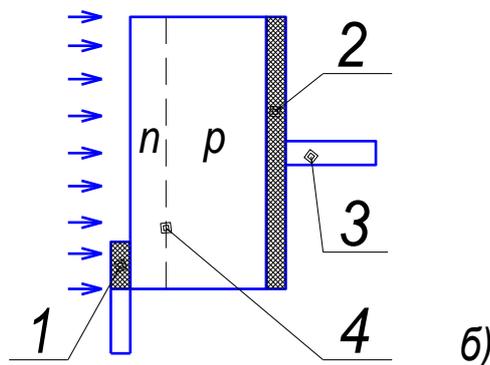


Рисунок 1.2.3 - Принципиальная схема фотодиода

Если внутри полупроводника граничат две области с разным типом проводимости, то возникает диффузия основных носителей тока: диффузионные токи электронов из n – области в p – область (ток I_{nn}) и дырок из p – области в n – область (ток I_{pp}). Приконтактные области объединяются основными носителями. Это приводит к появлению объемных зарядов вблизи границы, образованных неподвижными зарядами ионизированных атомов примеси (рисунок 1.2.4).

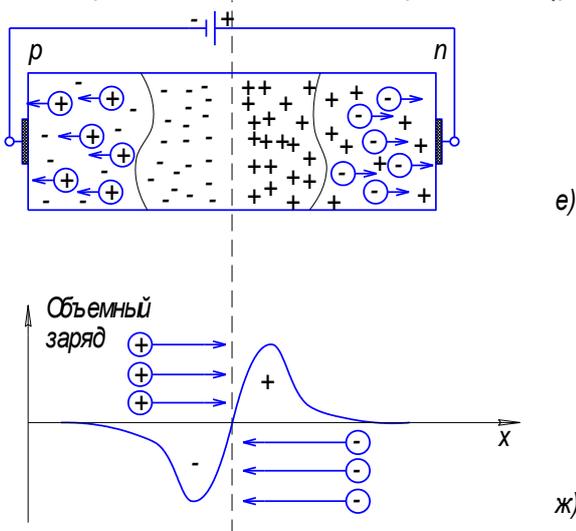


Рисунок 1.2.4 – Формирование p-n перехода

По мере нарастания объемных зарядов нарастает электрическое поле, противодействующее диффузии основных носителей (возникает потенциальный

барьер движению основных носителей и диффузионные токи основных носителей уменьшаются. Разность потенциалов этого поля называют контактной разностью потенциалов V_k , знак ее соответствует обеднению приконтактных областей основными носителями тока. Преодолеть потенциальный барьер могут только те из основных носителей, энергия которых больше энергии потенциального барьера ($E_k = eV_k$, e – заряд электрона).

Одновременно с диффузионным током основных носителей возникает встречный дрейфовый ток неосновных, т.к. для них контактное электрическое поле является ускоряющим. Дырки из n – области переходят в p – область (ток I_{pn}), а электроны из p – области – в n – область (ток I_{np}). Контактная разность потенциалов возрастает до тех пор, пока потоки основных и неосновных носителей через p - n переход не достигнут динамического равновесия:

$$I_{nn} = I_{np}; \quad I_{pp} = I_{pn} \quad (1.1.1)$$

При этом во внешней цепи ток отсутствует:

$$I_1 = I_{nn} + I_{pp} - I_{np} - I_{pn} = 0 \quad (1.1.2)$$

При приложении внешнего напряжения V_D потенциальный барьер изменяется на величину eV_D . Равновесие тока нарушается. При этом поток неосновных носителей через p - n переход изменяется незначительно, а ток основных носителей зависит от V_D : при прямом включении (плюс к p – области, минус к n – области) контактная разность потенциалов уменьшается и ток основных носителей очень быстро возрастает с увеличением V_D . При обратном включении контактная разность потенциалов увеличивается и ток основных носителей практически прекращается. Вольт-амперная характеристика неосвещенного p - n перехода приведена на рисунок 1.2.5 (кривая $\Phi = 0$), где за положительные приняты: запирающее напряжение на диоде и обратный ток диода.

При облучении одной из областей излучением с энергией квантов E_Φ , превышающей ширину запрещенной зоны собственного полупроводникового материала этой области (ΔE) – $E_\Phi > \Delta E$ – в объеме полупроводника генерируются пары неравновесных носителей тока – фотоэлектроны и фотодырки (внутренний фотоэффект в собственном полупроводнике). Когда диффундирующие в объеме полупроводника фотоносители достигают области p - n перехода, в контактом электрическом поле происходит пространственное разделение пар: основные фотоносители остаются в объеме той области, где они возникли; неосновные свободно проходят через p - n переход, так как для них контактное поле является ускоряющим. Таким образом, неосновные фотоносители создают внутри p - n перехода дополнительный ток, который называют фототоком I_f .

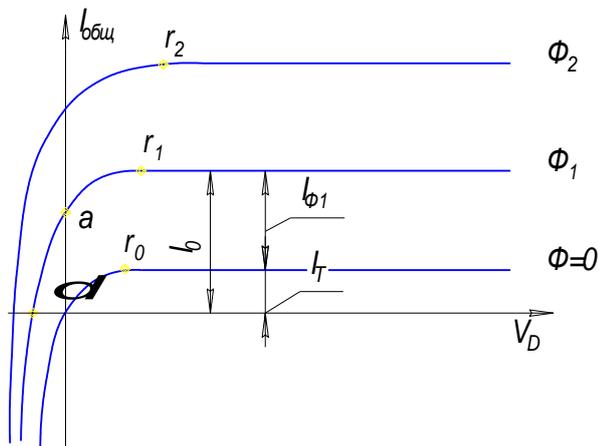


Рисунок 1.2.5 - Вольт-амперная характеристика $p-n$ перехода

В зависимости от способа включения фотодиода и способа измерения фотосигналом цепи включения может служить или фототок или напряжение фотосигнала.

Фототок фотодиода преобразуется в напряжение фотосигнала посредством включения в электрическую цепь сопротивления нагрузки R_H

Применяют два способа (основных) включения фотодиода: фотодиодный режим (рисунок 1.2.6(а)), вентильный режим (рисунок 1.2.6(б))

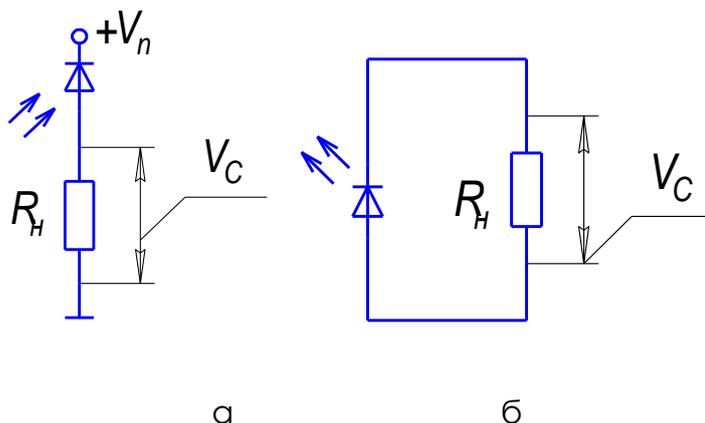


Рисунок 1.2.6 - Фотодиодный (а) и вентильный (б) режимы работы фотодиода

Вентильный режим характерен отсутствием внешнего источника питания, т.е. используется способность $p-n$ перехода фотодиода генерировать фотосигналы – фототок или фото ЭДС. В фотодиодном режиме внешнее напряжение V_D прикладывается в запирающем направлении.

3. Фоторезистор – это датчик, электрическое сопротивление которого меняется в зависимости от интенсивности падающего на него света. Чем интенсивней свет, тем больше создается свободных носителей зарядов и тем меньше становится сопротивление элемента.

Фоторезистор представляет собой тонкую пластинку или пленку полупроводника 1 с омическими контактами 2 на двух противоположных концах (рисунок 1.2.7), к которым подключается электрический источник питания. Полупроводник обычно наносится на стеклянную подложку 3.

Наиболее распространены фоторезисторы на основе CdS и $CdSe$, спектральные характеристики которых располагаются в видимой области спектра. В Инфракрасном диапазоне работают фоторезисторы из Ge , Si и т.д.

Падающее на поверхность фоторезистора излучение генерирует в нем свободные носители за счет собственного или примесного поглощения. Принцип

работы фоторезистора приведен на рисунке 1.2.8.

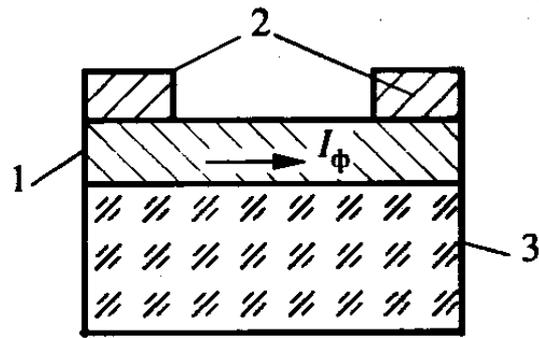
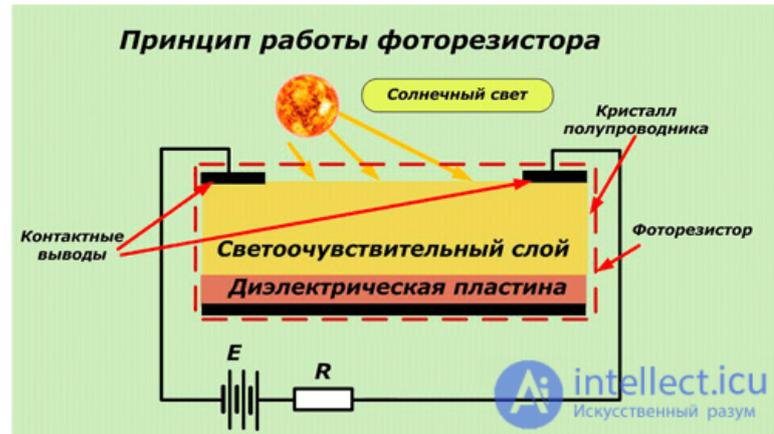


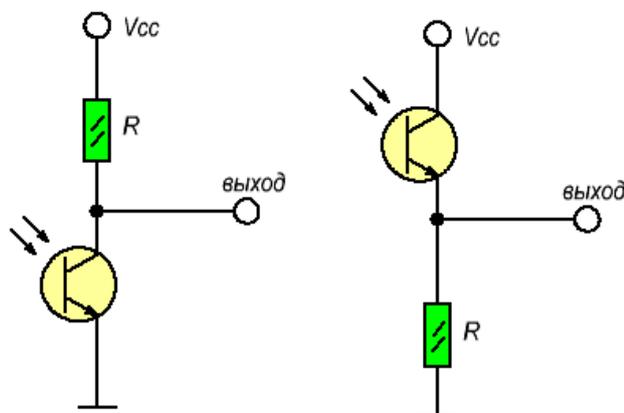
Рисунок 1.2.7 – Схема фоторезистора



1.2.8 – Принцип работы фоторезистора

4. Фототранзистор представляет собой твердотельное полупроводниковое устройство с внутренним усилением, которое используется для обеспечения аналоговых и цифровых сигналов. Фототранзисторы используются в детекторах дыма, лазерных радарах, системах дистанционного управления. Фототранзисторы более чувствительны по сравнению с фотодиодами, а также создают больший фототок.

Существует 2 основные схемы включения фототранзистора (рисунок 1.2.9).



1.2.9 – Схемы включения NPN транзистора. а – с общим эмиттером,
б – с общим коллектором

В случае включения по рисунку 1.2.9(а) сигнал переходит из высокого состояния в низкое в момент освещения фототранзистора, в случае 1.2.9(б) происходит обратное.

В обоих случаях фототранзисторы могут использоваться в двух режимах работы: работа в активном режиме, работа в режиме переключения.

Работа в активном режиме подразумевает, что фототранзистор генерирует выходной сигнал пропорциональный степени его освещенности. То есть значение выходного сигнала имеет обратную зависимость от освещенности. Работа в режиме переключения подразумевает, что фототранзистор в ответ на его освещение будет либо «выключен», либо «включен». Активный режим используется тогда, когда необходимо различать несколько порогов напряжения. Режим переключения используется тогда, когда необходимо получить цифровой выходной сигнал. Переход от одного режима к другому осуществляется подбором резистора (в цепи коллектора в случае 1.2.9(а) или эмиттера в случае 1.2.9(б)). Обычно для режима переключения используется резистор номиналом более 5кОм, а для активного режима менее 5кОм.

Структура

1.3. Порядок проведения экспериментов:

В работе используются:

- модули: «Мультиметры», «Датчики цвета и света», «Регулируемый источник питания постоянного тока».

1. Изучение датчика цвета.

1.1 Ознакомиться с схемой соединения датчика с измерительным прибором (рисунок 1.3.1). В качестве измерительных приборов используются измерительные приборы модуля «Мультиметры».



Рисунок 1.3.1 – Подключение мультиметра для исследования датчика цвета

1.2 Собрать схему проведения эксперимента по исследованию датчика.

1.3 Включить питание модулей.

1.4 Произвести исследование датчика без фильтра, с красным светофильтром, с зеленым светофильтром и синим светофильтром. Для этого произвести выбор светофильтра нажатием кнопки «ВЫБОР СВЕТОФИЛЬТРА» до загорания контрольного светодиода «ЦВЕТ» цветом, соответствующем выбранному светофильтру.

1.5 Поочередно изменяя цвета полос, измерить выходное напряжение каждого датчика с различными светофильтрами.

1.6 Данные занести в таблицу 1.3.1.

Таблица 1.3.1

Фильтр\Цвет полосы	красный	синий	зеленый
U _{вых} (мВ) без фильтра			
U _{вых} (мВ) красный			
U _{вых} (мВ) синий			
U _{вых} (мВ) зеленый			

1.7 Сделать соответствующие выводы по данному разделу лабораторной работы.

2. Изучение фотодиода и влияния силы света на его характеристики

2.1 Ознакомьтесь с схемой соединения фотодиода с измерительными приборами и источником питания (рисунок 1.3.2). Вольтметр необходимо установить в положение – измерение постоянного напряжения 20В. Амперметр необходимо установить в положение – измерение постоянного тока 2 мА.

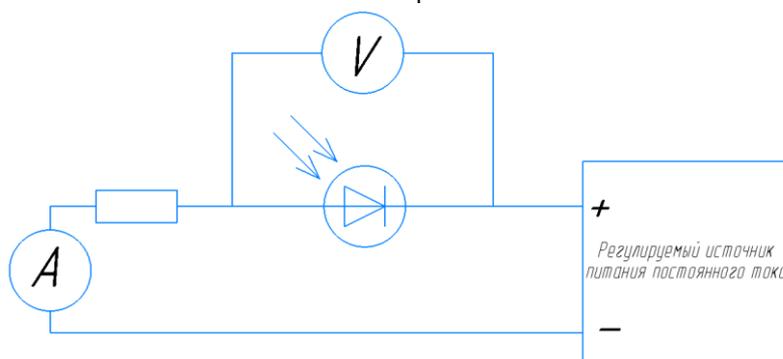


Рисунок 1.3.2 – Подключение мультиметра и источника питания для исследования характеристик фотодиода в фотодиодном режиме работы

2.2 Собрать схему проведения эксперимента по исследованию характеристик фотодиода.

2.3 Включить питание модулей.

2.4 Установить напряжение питания 3В на модуле «Регулируемый источник питания постоянного тока».

2.5 Изменяя значение силы света ($E(lx)$) при помощи ручки, зафиксировать значения напряжения и тока на мультиметрах и записать их в таблицу 1.3.2. В таблице указано значение освещенности для 11 рисок на поворотной ручке в порядке возрастания (0 соответствует 0 Lx, 1 соответствует 2150 Lx).

Таблица 1.3.2

Хар-ки\E(lx)	0	150	320	630	850	1100	1350	1610	1860	1990	2150
U _d											
I _d											

2.6 На основании полученных значений постройте зависимость фототока от освещенности.

2.7 Измените значение напряжения питания на 5В и повторите пункты 2.5 и 2.6.

2.8 Измените значение напряжения питания на 7В и повторите пункты 2.5 и 2.6.

2.9 Провести аналогичные опыты без источника питания (схема на рисунке 1.3.3).

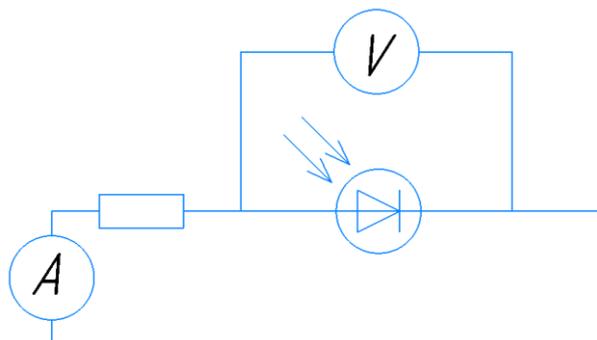


Рисунок 1.3.3 – Подключение мультиметра и источника питания для исследования характеристик фотодиода в вентильном режиме работы

2.10 Сделать соответствующие выводы по данному разделу лабораторной работы.

3. Изучение фоторезистора и влияния силы света на его характеристики

3.1 Ознакомьтесь с схемой соединения фоторезистора с измерительными приборами и источником питания (рисунок 1.3.4). Вольтметр необходимо установить в положение – измерение постоянного напряжения 20В. Амперметр необходимо установить в положение – измерение постоянного тока 20 мА.

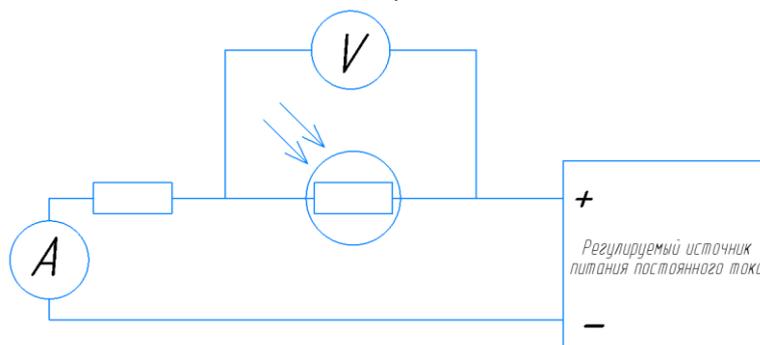


Рисунок 1.3.4 – Подключение мультиметра и источника питания для исследования характеристик фоторезистора

3.2 Собрать схему проведения эксперимента по исследованию характеристик фоторезистора.

3.3 Включить питание модулей.

3.4 Установить напряжение питания 3В на модуле «Регулируемый источник питания постоянного тока».

3.5 Изменяя значение силы света ($E(lx)$) при помощи ручки, зафиксировать значения напряжения и тока на мультиметрах и записать их в таблицу 1.3.3. В таблице указано значение освещенности для 11 рисок на поворотной ручке в порядке возрастания (0 соответствует 0 Lx, 1 соответствует 2150 Lx).

Таблица 1.3.3

Хар-ки \ $E(lx)$	0	150	320	630	850	1100	1350	1610	1860	1990	2150
U_{fr}											
I_{fr}											

3.6 На основании полученных значений постройте зависимость фототока от освещенности.

3.7 Измените значение напряжения питания на 5В и повторите пункты 3.5 и 3.6.

3.8 Измените значение напряжения питания на 7В и повторите пункты 3.5 и 3.6.

3.9 Сделать соответствующие выводы по данному разделу лабораторной работы.

4. Изучение фототранзистора и влияния силы света на его характеристики

4.1 Ознакомьтесь с схемой соединения фототранзистора с измерительными приборами и источником питания (рисунок 1.3.5). Вольтметр необходимо установить в положение – измерение постоянного напряжения 20В. Амперметр необходимо установить в положение – измерение постоянного тока 20 мА.

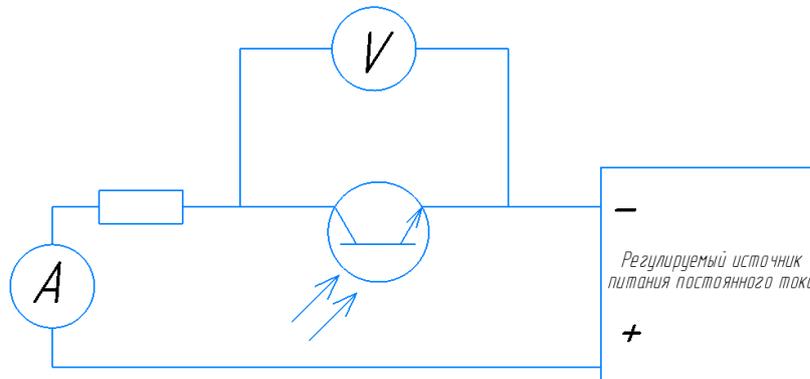


Рисунок 1.3.5 – Подключение мультиметра и источника питания для исследования характеристик фототранзистора в активном режиме

4.2 Собрать схему проведения эксперимента по исследованию характеристик фоторезистора.

4.3 Включить питание модулей.

4.4 Установить напряжение питания 3В на модуле «Регулируемый источник питания постоянного тока».

4.5 Изменяя значение силы света ($E(lx)$) при помощи ручки, зафиксировать значения напряжения и тока на мультиметрах и записать их в таблицу 1.3.4. В таблице указано значение освещенности для 11 рисок на поворотной ручке в порядке возрастания (0 соответствует 0 Lx, 1 соответствует 2150 Lx).

Таблица 1.3.4

Хар-ки \ $E(lx)$	0	150	320	630	850	1100	1350	1610	1860	1990	2150
U_{vt}											
I_{vt}											

4.6 На основании полученных значений постройте зависимость фототока от освещенности.

4.7 Измените значение напряжения питания на 5В и повторите пункты 4.5 и 4.6.

4.8 Измените значение напряжения питания на 7В и повторите пункты 4.5 и 4.6.

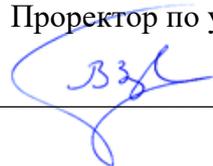
4.9 Сделать соответствующие выводы по данному разделу лабораторной работы, определить режим работы фототранзистора.

ЗАМЕТКИ

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»

Проректор по учебно-методической
работе
В. В. Зубов



Методические указания к выполнению лабораторных работ

Б1.В.09.07 ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Специальность -

21.05.04 Горное дело

Специализация -

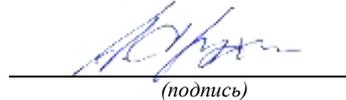
Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Информатики

(название кафедры)

Зав. кафедрой



(подпись)

Дружинин А.В.

(Фамилия И.О.)

Протокол № 1 от 19.09.2024

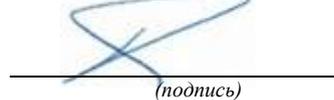
(Дата)

Рассмотрен методической комиссией
факультета

Горно-механического

(название факультета)

Председатель



(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

Введение

Цель данных лабораторных работ состоит в закреплении теоретических основ курса «Интеллектуальные системы» для студентов специальности «Прикладная математика», включающего изучение семейства методов оптимизации, известных как генетические алгоритмы.

Все работы проводятся в лаборатории, оборудованной персональными компьютерами под управлением *Microsoft Windows XP* или выше. Рекомендуемой средой для решения задач является *MATLAB*. Студенты, впервые приступившие к работе, обязаны пройти инструктаж по технике безопасности и расписаться в соответствующем журнале. К выполнению лабораторных работ допускаются только студенты, получившие допуск преподавателя после собеседования о теоретической части и основах работы в среде *MATLAB*.

Порядок выполнения лабораторных работ

1. Загрузить систему *MATLAB* и изучить справочные материалы по функциям системы, необходимым для решения задачи.
2. Составить алгоритм и написать программу на языке *MATLAB* для решения задачи.
3. Оформить отчет по лабораторной работе. Каждый отчет должен содержать следующие элементы.
 - 3.1. Формулировку задачи и числовые данные конкретных примеров.
 - 3.2. Текст программы.
 - 3.3. Результаты работы программы.

Лабораторная работа №1. Оптимизация функций вещественных переменных

Цель работы

Приобретение навыков работы со стандартным модулем *MATLAB*, предоставляющим возможность решения различных оптимизационных задач с помощью генетических алгоритмов.

Методические указания

Генетические алгоритмы (ГА) относятся к семейству алгоритмов глобальной оптимизации. Первоначальная форма ГА, введенная Дж. Холландом, является алгоритмом перехода от рассматриваемой по-

пуляции хромосом¹, под которыми понимались последовательности бит фиксированной длины, к новой популяции с помощью своего рода естественного отбора, сопровождаемого применением так называемых операторов *скрещивания, мутации и инверсии*. Оператор селекции выбирает хромосомы, которым будет позволено воспроизводиться, так чтобы хромосомы с лучшими значениями целевой функции в среднем имели большее потомство. Оператор скрещивания меняет местами соответствующие отрезки двух хромосом, имитируя биологическую рекомбинацию; оператор мутации инвертирует значения случайным образом выбранных бит хромосомы; оператор инверсии переворачивает непрерывный отрезок хромосомы.

Впоследствии интенсивное взаимодействие между исследователями, изучающими различные эволюционные методы, привело к тому, что границы между ГА, эволюционными стратегиями, эволюционным программированием и другими подходами такого рода стали несколько размыты. Вследствие этого термин «генетический алгоритм» нередко обозначает нечто весьма отличное от оригинальной формулировки, данной Холландом. Таким образом, под генетическим алгоритмом понимается не столько конкретный алгоритм, сколько целое семейство методов, объединенных общей концепцией.

В целом, ГА описывает способ эволюционирования популяции, состоящей из множества индивидуумов, по определенным правилам таким образом, чтобы собрать как можно больше индивидуумов в области с как можно лучшими значениями целевой функции. Общая структура ГА включает следующие шаги.

1. Создается исходная популяция, обычно случайная.
2. Для каждого индивидуума вычисляется значение целевой функции.
3. К популяции применяются операторы селекции, скрещивания и мутации. В результате получается новая популяция такого же размера.
 - 3.1. Оператор селекции выбирает индивидуумов, которые будут участвовать в процессе репродукции. При этом вероятность выбора каждого конкретного индивидуума тем выше, чем лучше его значение целевой функции. Индивидуум может как быть выбран несколько раз, так и не быть выбран вовсе.

¹ Правильнее было бы *популяции особей*, но в ГА не делается различия между особями и их хромосомами.

- 3.2. Оператор скрещивания формирует одного или большее количество потомков, комбинируя хромосомы двух родителей (в некоторых случаях может использоваться большее количество родительских индивидуумов).
- 3.3. Оператор мутации модифицирует хромосому индивидуума случайным образом. Он может использоваться после оператора скрещивания, или же вместо него для генерации потомков с использованием хромосомы единственного родителя.
4. Текущая популяция заменяется новой.
5. Процесс повторяется с шага 2, пока не будет выполнен оговоренный заранее критерий останова.

В качестве хромосом/индивидуумов могут применяться объекты самого разного типа: битовые, целочисленные, вещественные или комплексные векторы, конечные автоматы, деревья выражений и пр.

В состав системы *MATLAB*² включен модуль «*Genetic Algorithm and Direct Search*» (*GADS*). Помимо ГА, в него также входят так называемые алгоритмы прямого поиска (*Direct/Pattern Search*) и имитации отжига (*Simulated Annealing and Threshold Acceptance*). Для работы с ГА предназначены следующие функции: *ga*, *gaoptimget* и *gaoptimset*, *gatool*. Первая является собственно реализацией ГА, две следующие предназначены для манипулирования ее параметрами, а последняя вызывает графический интерфейс, имеющий идентичную функциональность, но большую наглядность. Приведем несколько примеров использования данных функций.

Простейший пример: поиск минимумов функций одной переменной x^2 и $\sin x$ (поскольку ГА имеет стохастическую природу, при каждом его вызове ответы получаются различные).

```
>> ga(@(x)x^2,1)
ans =
    0.0048
>> ga(@sin,1)
ans =
   -1.5650
```

Не очень высокая точность результатов является следствием щадящих значений параметров ГА, выставляемых по умолчанию. Интересной особенностью второго примера является то, что минимум найден вне

² Приведенные здесь сведения относятся к *GADS* версии 2.1 в составе *MATLAB* 7.4 (R2007a).

начального диапазона поиска (по умолчанию отрезок $[0; 1]$): в общем случае, в зависимости от параметров алгоритма, популяция в поисках экстремума может уйти довольно далеко от своего изначального месторасположения. Тем не менее, с точки зрения вычислительной эффективности выгоднее сразу нацеливать ГА на область, в которой, предположительно, находится решение.

Аналогично производится поиск минимума функции большего количества переменных (здесь $f(x) = (\sin x_1)(\sin x_2)$).

```
>> ga(@(x)sin(x(1))*sin(x(2)),2)
ans =
    -1.5663    1.5724
```

Чтобы получить более адекватный ответ, воспользуемся возможностью уточнения результатов с помощью традиционных алгоритмов минимизации (в данном случае стандартной функции `fminsearch`).

```
>> opt=gaoptimset('HybridFcn',@fminsearch);
>> ga(@(x)sin(x(1))*sin(x(2)),2,...
    [],[],[],[],[],[],[],opt)
ans =
    -1.5708    1.5708
```

Здесь `HybridFcn` — один из множества параметров, с помощью которых универсальный алгоритм, представленный функцией `ga`, адаптируется под конкретные задачи.

Общий синтаксис `ga` выглядит следующим образом:

```
x = ga(fitnessfcn,nvars,A,b,...
    Aeq,beq,LB,UB,nonlcon,options)
x = ga(problem)
[x,fval,exitflag,output,population,scores] = ga(...)
```

Здесь `fitnessfcn` — целевая функция с одним (векторным) аргументом; `nvars` — количество неизвестных. Параметры `A` и `b` описывают линейные ограничения в виде неравенств: $Ax \leq b$; `Aeq` и `beq` — линейные ограничения в виде равенств: $A_{eq}x = b_{eq}$. Параметры `LB` и `UB` представляют собой нижние и верхние границы значений неизвестных (*Lower Bounds* и *Upper Bounds* соответственно): $x_i \in [LB_i, UB_i]$. Параметр `nonlcon` — функция, описывающая нелинейные ограничения задачи. Эта функция принимает в качестве аргумента `x` и возвращает два вектора `C` и `Seq`; ГА, в свою очередь, ищет такое решение задачи, чтобы вы-

полнялись условия $C \leq 0$ и $C_{eq} = 0$. Последний аргумент — структура, формируемая с помощью функции `gaoptimset`. Все аргументы, отвечающие за ограничения, имеют смысл только для индивидуумов типа `doubleVector`. Обязательными в первой форме вызова являются лишь первые два аргумента; остальные можно опустить или указать как `[]`. Во второй форме вызова единственный аргумент является структурой, формируемой при экспорте задачи из интерфейса `gatool`.

На выходе функции имеем вектор `x`, дающий искомую точку минимума; значение целевой функции в этой точке `fval`; числовой код `exitflag`, указывающий причину прекращения работы ГА (коды см. в справке); структуру `output`, содержащую вспомогательную информацию, которая, в частности, может применяться для воспроизведения работы алгоритма с сохранением идентичных результатов; финальную популяцию `population` и значения целевой функции на ней `scores`. Все выходные параметры (кроме `x`) опциональны.

Варианты задания

Построить графики (поверхности и линии уровня) следующих тестовых функций для размерности пространства поиска $n = 2$. Используя командный (`ga`) и графический интерфейсы (`gatool`) модуля *GADS*, продемонстрировать нахождение их минимумов для $n = 2, 5, 10, 100$ (где это применимо).

Вар.	Функция
1.	<p>Сферическая функция, или функция Де Джонга 1 (<i>sphere model, De Jong's function 1</i>) — гладкая выпуклая унимодальная функция, мечта любого алгоритма оптимизации.</p> $f(x) = \sum_{i=1}^n x_i^2, \quad x_i \in [-5,12; 5,12].$ <p>(Глобальный минимум: $x_i = 0$.)</p>
2.	<p>Гиперэллипсоидная функция (<i>axis parallel hyper-ellipsoid function</i>).</p> $f(x) = \sum_{i=1}^n ix_i^2, \quad x_i \in [-5,12; 5,12].$ <p>(Глобальный минимум: $x_i = 0$.)</p>
3.	<p>Повернутая гиперэллипсоидная функция (<i>rotated hyper-ellipsoid function</i>).</p>

	$f(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2, \quad x_i \in [-65,536; 65,536].$ <p>(Глобальный минимум: $x_i = 0$.)</p>
4.	<p>Ступенчатая функция, или функция Де Джонга 3 (<i>step function, De Jong's function 3</i>) состоит из множества плоских горизонтальных поверхностей. Подобные функции создают проблемы многим алгоритмам оптимизации, поскольку такие плоскости не дают информации о предпочтительном направлении поиска.</p> $f(x) = 6n + \sum_{i=1}^n [x_i] , \quad x_i \in [-5,12; 5,12].$
5.	<p>Функция (она же банановая функция, она же долина) Розенброка, или функция Де Джонга 2, (<i>Rosenbrock's function (banana function, valley), De Jong's function 2</i>). Классическая задача о поиске экстремума, находящегося в узкой изогнутой долине. Найти последнюю легко, но спуск по ней к минимуму — задача нетривиальная. Функция является мультимодальной при $n > 3$.</p> $f(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right), \quad x \in [-5,12; 5,12].$ <p>(Глобальный минимум: $x_i = 1$.)</p>
6.	<p>Функция Де Джонга с гауссовыми возмущениями (<i>gaussian quartic, De Jong's function 4</i>) позволяет оценить надежность алгоритма оптимизации при работе с зашумленными данными.</p> $f(x) = \sum_{i=1}^n \left(ix_i^4 + gauss(0,1) \right), \quad x_i \in [-1,28; 1,28],$ <p>где $gauss(m, d)$ — функция, возвращающая нормально распределенные случайные величины с математическим ожиданием m и дисперсией d.</p>
7.	<p>Функция Растригина (<i>Rastrigin's function</i>): аналог функции №1 с многочисленными регулярно распределенными минимумами.</p> $f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos 2\pi x_i), \quad x_i \in [-5,12; 5,12].$ <p>(Глобальный минимум: $x_i = 0$.)</p>
8.	<p>Функция Швевеля (<i>Schwefel's sine root function</i>) имеет обманчивый рельеф: глобальный минимум находится вдалеке от локального ми-</p>

	<p>нимума, наиболее схожего с ним по величине функции.</p> $f(x) = \sum_{i=1}^n \left(-x_i \sin \sqrt{ x_i } \right), \quad x_i \in [-500; 500].$ <p>(Глобальный минимум: $x_i = 420,9687$.)</p>
9.	<p>Функция Грайвенка (<i>Griewangk's function</i>). Мультимодальная функция с регулярно распределенными минимумами; аналог функции Растригина.</p> $f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right), \quad x_i \in [-600; 600].$ <p>(Глобальный минимум: $x_i = 0$.)</p>
10.	<p>Функция Эккли (<i>Ackley's path function</i>): часто используемая мультимодальная тестовая функция.</p> $f(x) = 20 + e - 20 \exp \left(-0,2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right),$ <p>$x_i \in [-30; 30]$.</p> <p>(Глобальный минимум: $x_i = 0$.)</p>
11.	<p>Сумма различных степеней (<i>Sum of different powers</i>): часто используемая унимодальная тестовая функция.</p> $f(x) = \sum_{i=1}^n x_i ^{i+1}, \quad x_i \in [-1; 1].$ <p>(Глобальный минимум: $x_i = 0$.)</p>
12.	<p>Функция Михалевича (<i>Michalewicz's function</i>): мультимодальная функция с $n!$ локальными экстремумами, рельеф которой состоит из почти плоских участков с глубокими прямыми расщелинами. При больших n последние становятся уже, их склоны — вертикальнее, а задача оптимизации превращается в своего рода поиск иголки в стоге сена.</p> $f(x) = - \sum_{i=1}^n \left(\sin(x_i) \sin^{2m} \left(\frac{i x_i^2}{\pi} \right) \right), \quad m = 10, x_i \in [0; \pi].$
13.	<p>Функция Бранинса (<i>Branins's rcos function</i>): тестовая функция с тремя глобальными минимумами.</p> $f(x_1, x_2) = \left(x_2 - \frac{5,1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 +$

	$+10 \left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10, \quad x_1 \in [-5; 10], x_2 \in [0; 15].$ <p>(Глобальные минимумы: $(-\pi; 12,275)$, $(\pi; 2,275)$, $(9,425; 2,475)$.)</p>
14.	<p>Функция Изома (<i>Easom's function</i>): унимодальная тестовая функция, в которой глобальный минимум и окружающий его нетривиальный рельеф занимают лишь небольшую часть пространства поиска.</p> $f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2),$ $x_{1,2} \in [-100; 100].$ <p>(Глобальный минимум: $(\pi; \pi)$.)</p>
15.	<p>Функция Голдстейна (<i>Goldstein-Price function</i>):</p> $f(x_1, x_2) =$ $= [1 + (1 + x_1 + x_2)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$ $\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)];$ $x_{1,2} \in [-2; 2].$ <p>(Глобальный минимум: $(0; -1)$.)</p>
16.	<p>Шестигорбая функция (<i>Six-hump camel back function</i>) имеет в области поиска шесть локальных минимумов, два из которых являются глобальными.</p> $f(x_1, x_2) = \left(4 - 2,1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2;$ $x_1 \in [-3; 3], x_2 \in [-2; 2].$ <p>(Глобальные минимумы: $(-0,0898; 0,7126)$, $(0,0898; -0,7126)$.)</p>

Вопросы для собеседования

1. Что такое генетический алгоритм?
2. Какова общая схема работы ГА?
3. Как производится решение задач с помощью ГА в *MATLAB*?
4. Чем отличается реализация ГА в *MATLAB* от канонического алгоритма?

Лабораторная работа №2. Оптимизация функций вещественных переменных, часть 2

Цель работы

Приобретение навыков реализации и применения генетических алгоритмов в задачах оптимизации функций нескольких вещественных переменных.

Методические указания

Существует большое количество разновидностей операторов, соответствующих шагу 3 общего ГА. Выбор тех или иных вариантов определяется в первую очередь типом хромосом. Приведем варианты операторов, применимых в задачах оптимизации функций вещественных переменных.

1. Селекция (для поиска минимума).
 - 1.1. Популяция упорядочивается по значениям целевой функции, после чего ее вторая половина отбрасывается, а первая дублируется.
 - 1.2. Популяция упорядочивается по значениям целевой функции. Каждому индивидууму назначается вес, равный $1/\sqrt{n}$, где n — номер его позиции. Производится случайный выбор индивидуума, причем вероятность выбора каждого из индивидуумов пропорциональна его весу. Процесс повторяется столько раз, сколько особей должно быть в популяции.
 - 1.3. Выбираются несколько случайных индивидуумов. Лучший из них помещается в новую популяцию. Процесс повторяется столько раз, сколько особей нужно выбрать.
2. Скрещивание. Популяция перемешивается случайным образом, после чего к первым $[nr_c/2]$ парам индивидуумов x_{2i-1} и x_{2i} , $i = 1, \dots, [nr_c/2]$, где n — размер популяции, а r_c — вероятность скрещивания, применяется один из следующих вариантов действий.
 - 2.1. Производится одноточечный или двухточечный кроссовер векторов-хромосом.
 - 2.2. Составляются (масштабированные) выпуклые линейные комбинации родительских хромосом.
 - 2.3. Симулированное бинарное скрещивание:

$$x'_{1,2} = ((1 \pm \beta)x_1 + (1 \mp \beta)x_2)/2,$$

$$\beta = \begin{cases} (2u)^{1/(\eta+1)}, & u \leq 1/2, \\ (2(1-u))^{-1/(\eta+1)}, & u > 1/2, \end{cases}$$

где $x_{1,2}$ — родительские хромосомы, $x'_{1,2}$ — потомки, u — случайная величина, равномерно распределенная на $[0; 1]$, а η — выбираемый пользователем параметр, определяющий форму функции распределения коэффициента β .

3. Мутация. К $[nr_m]$ случайно выбранным индивидуумам, где n — размер популяции, а r_m — вероятность мутации, применяется один из следующих вариантов действий.
 - 3.1. К выбранным векторам прибавляются нормально распределенные смещения с нулевым математическим ожиданием и выбранной заранее дисперсией.
 - 3.2. Каждая координата выбранных векторов с некоторой вероятностью заменяется на случайное число из диапазона, соответствующего области поиска.

Варианты задания

Написать программу, реализующую генетический алгоритм, и решить с ее помощью задачу на нахождение минимума функции с одним экстремумом, с несколькими локальными экстремумами, а также минимума линейной функции на выпуклом множестве:

- $f(x, y) = (x - 0,4)^2 + (y - 0,6)^4 \rightarrow \min;$
- $g(x, y) = f(x, y) + (\sin 10x)(\sin 10y) \rightarrow \min;$
- $\begin{cases} x + y \rightarrow \min, \\ f(x, y) \leq 1. \end{cases}$

На каждой итерации ГА показывать текущую популяцию в виде облака точек. Программы для каждого из трех пунктов должны отличаться *только* видом целевой функции.

Сравнить результаты с (более) точными, найденными аналитически или с помощью функций, реализующих традиционные алгоритмы оптимизации.

Решить те же задачи с помощью стандартного модуля *MATLAB GADS*. Сравнить количество итераций и время, необходимое для достижения аналогичной точности результата.

Ввиду особенностей языка *MATLAB* программа должна использовать технику векторизации вычислений везде, где это возможно, для достижения высокого быстродействия.

Операторы селекции, скрещивания и мутации выбираются из приведенного выше списка в соответствии со своим вариантом (параметры операторов оставлены на усмотрение разработчика).

Вариант	Селекция	Скрещивание	Мутация
1	1.1.	2.1.	3.1.
2	1.1.	2.1.	3.2.
3	1.1.	2.2.	3.1.
4	1.1.	2.2.	3.2.
5	1.1.	2.3.	3.1.
6	1.1.	2.3.	3.2.
7	1.2.	2.1.	3.1.
8	1.2.	2.1.	3.2.
9	1.2.	2.2.	3.1.
10	1.2.	2.2.	3.2.
11	1.2.	2.3.	3.1.
12	1.2.	2.3.	3.2.
13	1.3.	2.1.	3.1.
14	1.3.	2.1.	3.2.
15	1.3.	2.2.	3.1.
16	1.3.	2.2.	3.2.
17	1.3.	2.3.	3.1.
18	1.3.	2.3.	3.2.

Вопросы для собеседования

1. Приведите примеры операторов селекции.
2. Приведите примеры операторов скрещивания и мутации, используемых в дискретных ГА.
3. Приведите примеры операторов скрещивания и мутации, используемых в непрерывных ГА.

4. Почему недостаточно одной лишь классической (бинарной) версии ГА?

Лабораторная работа №3. Эволюционный процесс

Цель работы

Применение генетических алгоритмов к оптимизационным задачам нестандартного вида.

Методические указания

ГА обладают рядом преимуществ перед традиционными методами оптимизации, в частности:

- переменные целевой функции могут быть как непрерывными, так и дискретными;
- отсутствует необходимость в вычислении производных;
- производится одновременный поиск в широком диапазоне значений переменных;
- возможна эффективная работа с функциями большого числа переменных;
- возможно эффективное распараллеливание алгоритмов;
- ГА применимы к целевым функциям со сложным рельефом, с большим количеством локальных экстремумов;
- результат может представлять собой перечень возможных оптимальных решений, а не просто одну точку в пространстве поиска;
- могут применяться различные формы представления (кодирования) переменных;
- целевые функции могут иметь самую разную природу: замкнутые формулы, вычисления на основе сложных алгоритмов, экспериментальные данные и пр.

Данные свойства позволяют ГА решать задачи, не поддающиеся традиционным методам. Более того, существуют вариации³, в рамках которых индивидуумы не просто указывают на некоторую точку пространства поиска, а, в некотором смысле, «ведут активный образ жизни». Это эволюционное и генетическое программирование. Роль хромосом играют в первом случае конечные автоматы, а во втором — программы. В то время как в ГА целевая функция фиксирована, а индивидуумы представля-

³ Их не относят напрямую к ГА, но считают родственными алгоритмами.

ют собой *данные*, здесь ситуация обратная: исходные *данные* выбираются заранее, а популяция состоит из *функций*, представленных в специфическом виде. Общая схема алгоритма мало чем отличается от ГА. Наиболее существенная разница связана с вычислением целевой функции и операторами скрещивания и мутации.

Задача, поставленная в данной лабораторной работе, не сводится к оптимизации какой-либо конкретной функции вещественных переменных. Тем не менее, она естественным образом решается с помощью ГА, если позволить целевой функции ранжировать индивидуумов на основании не только их собственных характеристик, но также и характеристик остальных индивидуумов в популяции.

Таким образом, если в предыдущей работе целевая функция *могла* быть сконструирована так, чтобы применяться ко всей популяции сразу, здесь она *должна* быть написана именно таким образом. В качестве аргумента ей передается весь имеющийся набор хромосом, а значения функции формируются на основе сравнения нескольких (в данном случае двух) хромосом. При этом целевая функция фактически частично берет на себя роль оператора селекции.

Варианты задания

Смоделировать эволюционный процесс, определить его сходимость, локализовать устойчивое значение генотипа в $(0,05; 0,95)^2$ для целевой функции, назначающей каждой особи ранг путем случайного разбиения популяции на пары с последующим проведением поединков в каждой из них.

Две координаты в составе хромосом задают, соответственно, выносливость (x_i) и силу (y_i) особи. Кроме того, для каждой особи формулой $p_i = 1 - (x_i + y_i)/2$ задана вероятность попадания по противнику. В начале поединка каждая особь получает запас энергии, равный ее выносливости. В ходе поединка особи по очереди пытаются нанести друг другу удары (попадая со своей вероятностью p_i); в случае успеха энергия атакуемого уменьшается на величину, равную силе атакующего (p_i не изменяется). Как только энергия одной из особей становится неположительной, ее противник выигрывает поединок.

По итогам турниров ранги назначаются таким образом, чтобы проигравшие отсеивались, а победители оценивались по трем следующим вариантам критерия.

Вариант	Цель ранжирования победителей
1.	Без предпочтений.
2.	Минимизация количества раундов поединка.
3.	Максимизация количества раундов поединка.

Вопросы для собеседования

1. Какие понятия заимствует ГА из биологии?
2. Что такое векторизация целевой функции, для чего она применяется?
3. Каковы преимущества и недостатки ГА по сравнению с традиционными методами оптимизации?
4. Какие существуют разновидности ГА?
5. В каких еще, кроме ГА, задачах и методах оптимизации используются эволюционные аналогии?

Лабораторная работа №4. Рефлексивная метаоптимизация

Цель работы

Поиск параметров генетического алгоритма, обеспечивающих наилучшую его сходимость.

Методические указания

Генетические алгоритмы представляют собой не классические целостные последовательности действий, но системы взаимодействующих друг с другом компонент (популяция и модифицирующие ее операторы). Каждая из этих компонент может иметь различные качественные и количественные характеристики. Например, популяция может характеризоваться типом индивидуумов, их областью значений и количеством. Каждый оператор характеризуется своим алгоритмом и его числовыми параметрами, а скрещивание и мутация, кроме того, вероятностью применения.

В зависимости от перечисленных характеристик эффективность генетического алгоритма может существенно различаться. С самого начала развития технологии ГА предпринимались попытки поиска оптимальных параметров. В данной лабораторной работе предлагается один из возможных подходов к этой задаче, основанный на технике рефлексивной метаоптимизации.

При предложенном ниже выборе параметров, подвергаемых оптимизации, данная лабораторная работа может служить иллюстрацией к вопросу о целесообразности совместного применения операторов скрещивания и мутации. Поскольку они решают одну и ту же задачу (создание индивидуумов для нового поколения), может возникнуть желание исключить один из них, чтобы избавиться от дублирования функциональности.

Рассматривая данные операторы по отдельности, исследователи пришли к следующим выводам. Если целевая функция имеет высокую степень регулярности и четко выраженный глобальный минимум, использование оператора скрещивания позволяет существенно ускорить поиск последнего. При увеличении числа локальных минимумов эффективность оператора скрещивания существенно снижается. Аналогичный эффект имеет место, если значения целевой функции в точках локальных минимумов приближаются к величине глобального минимума. С другой стороны, при использовании исключительно оператора мутации на работу ГА практически не влияют ни количество локальных экстремумов, ни их величины.

Таким образом, взаимные преимущества двух данных видов операторов проявляются в различных классах задач. В целом, использование скрещивания позволяет ускорить сходимость ГА, в то время как мутация, внося в популяцию помехи, замедляет стягивание ее к искомой точке, одновременно защищая от вырождения (преждевременного схождения к ложным экстремумам).

Варианты задания

Взять решения трех подзадач лабораторной работы №2, соответствующие своему варианту. Выбрать 2–3 параметра ГА, подлежащие оптимизации (например, вероятность скрещивания, вероятность и силу мутации). Выбрать для них адекватные области поиска, а для остальных параметров — подходящие фиксированные значения. Сформировать целевую функцию верхнего уровня, которая принимает значения выбранных параметров в качестве аргументов, производит поиск решения соответствующим алгоритмом и возвращает некоторый показатель качества результата (на выбор разработчика), т.е. меру его близости к истинному положению минимума. Для каждой подзадачи определить положение минимума этой целевой функции с помощью ГА, сравнить и объяснить результаты.

Вопросы для собеседования

1. Какова функция оператора селекции?
2. Какова функция оператора скрещивания?
3. Какова функция оператора мутации?
4. Чем отличаются роли операторов скрещивания и мутации?
5. Что такое гибридный ГА?

Библиографический список

1. Андрейчиков А. В., Андрейчикова О. Н. Интеллектуальные информационные системы: Учебник. — М.: Финансы и статистика, 2004.
2. Афонин В. Л., Макушкин В. А. Интеллектуальные робототехнические системы: курс лекций: учеб. пособие для студентов вузов, обучающихся по специальностям в области информ. технологий. — М.: Интернет-Ун-т Информ. Технологий, 2005.
3. Букатова И. Л. и др. Эвоинформатика. Теория и практика эволюционного моделирования. — М.: Наука, 1991.
4. Поздяев В. В. Интеллектуальные системы. Уч. пособие. — Н. Новгород: Изд-во НГТУ, 2008.

Содержание

Введение.....	3
Порядок выполнения лабораторных работ.....	3
Лабораторная работа №1. Оптимизация функций вещественных переменных	3
Лабораторная работа №2. Оптимизация функций вещественных переменных, часть 2	11
Лабораторная работа №3. Эволюционный процесс.....	14
Лабораторная работа №4. Рефлексивная метаоптимизация	16
Библиографический список.....	18

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»

Проректор по учебно-методической
работе
В. В. Зубов



**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО САМОСТОЯТЕЛЬНОЙ
РАБОТЕ СТУДЕНТОВ**

Б1.В.09.07 ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Специальность -

21.05.04 Горное дело

Специализация -

Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Информатики

(название кафедры)

Зав. кафедрой



(подпись)

Дружинин А.В.

(Фамилия И.О.)

Протокол № 1 от 19.09.2024

(Дата)

Рассмотрен методической комиссией
факультета

Горно-механического

(название факультета)

Председатель



(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

ВВЕДЕНИЕ

Интеллектуальные информационные системы проникают во все сферы нашей жизни, поэтому становятся неотъемлемым элементом при решении задач автоматизации и управления сложными объектами и процессами.

Современное понятие интеллектуальных систем (ИС) сформировалось в процессе развития теоретических основ кибернетики, современной теории управления, теории алгоритмов, развития современных информационных технологий и обобщения накопленных научных знаний, методов и средств в области искусственного интеллекта (ИИ).

Целью текста лекций является обзорное ознакомление студентов, обучающихся по направлению «Информатика и вычислительная техника» и другим родственным направлениям, с проблематикой и областями использования искусственного интеллекта в автоматизированных системах обработки информации и управления, освещение теоретических и организационно-методических вопросов построения и функционирования систем, основанных на знаниях, привитие навыков практических работ по проектированию баз знаний. В результате изучения лекционного материала студенты получают знания по архитектуре и классификации ИС, методам представления знаний, областям применения, а также научатся выбирать адекватные проблемной области методы проектирования базы знаний.

Поэтому в соответствии с требованиями федерального государственного образовательного стандарта высшего профессионального образования для формирования у студентов общекультурных, общепрофессиональных и профессиональных компетенций в модуль «Профессиональные дисциплины» учебного плана подготовки магистров по направлению 09.04.01 «Информатика и вычислительная техника» введена учебная дисциплина «Интеллектуальные системы».

Изучение студентами дисциплины «Интеллектуальные системы» должно ознакомить с основными понятиями и концепциями теории интеллектуальных систем, а также осветить вопросы построения и функционирования ИС, основанных на знаниях, с целью вооружить будущих магистров теоретическими знаниями и практическими навыками, необходимыми для решения задач создания и использования современных интеллектуальных информационных технологий и систем в области информационно-аналитического обеспечения подготовки и принятия управленческих решений по всем аспектам экономических, социальных и технических проблем.

Предполагается, что студенты уже прослушали курсы по информатике, программированию и основам алгоритмизации, базам данных в информационно-управляющих системах, математическим основам кибернетики, теории систем и управления, моделированию систем, системам автоматизации и управления.

В связи с большим объемом информации при ограниченном количестве аудиторных часов в одном семестре (18 часов лекционных и 18 часов практических занятий) в определенной мере связаны ограниченный объем учебного по-

собия и краткость изложения материала, что во многом и предопределило его обзорный характер. Поэтому основная часть материала дана в виде краткой справочной информации со ссылками на более полные источники. В соответствии с этим в список рекомендуемой литературы включены источники, необходимые для углубленного изучения рассматриваемых вопросов. К настоящему времени издано множество научно-технической литературы по указанной проблематике, включая и учебно-методические издания из различных вузов. При формировании списка литературы учитывалась реальная возможность доступа студентов к рекомендуемым материалам, поэтому представленный список не претендует на полноту и не является исчерпывающим перечнем. В связи с этим ограничены ссылки на известные, но трудно доступные классические основополагающие работы, а в первую очередь указаны ссылки на более доступные современные книги, а также различные методические и справочные материалы в Интернете, отдельные фрагменты из которых частично использованы в данном пособии.

По причине ограниченности часов дисциплины внимание в тексте лекций уделено такому актуальному направлению ИИ как интеллектуальный анализ данных (ИАД). Приводится классификация методов и программных средств реализации ИАД, которые будут более подробно изучаться на практических занятиях.

Курс лекций включает 18 часов лекций, которые в соответствии с рабочей программой дисциплины сгруппированы в 3 раздела:

1. Введение в интеллектуальные системы.
2. Модели представления знаний. Способы формализации интеллектуальных задач с помощью методов искусственного интеллекта.
3. Интеллектуальный анализ данных. Методы ИАД и их программная реализация.

В конце каждого раздела помещены контрольные вопросы, которые служат для проверки качества усвоения и понимания материала и могут быть включены в экзаменационные билеты.

Фактическое содержание данного текста лекций является введением в научные и методические основы создания и применения современных интеллектуальных систем и технологий применительно к решению различных сложных задач управления в компаниях. Приведенный в тексте лекций материал интегрирует информацию из наиболее актуальных и значимых источников, что позволяет в лаконичной форме дать магистрантам результаты, достигнутые в теории и практике ИС.

Глава 1. ПОНЯТИЯ И ТЕРМИНОЛОГИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

1.1. История исследований и основные понятия в области искусственного интеллекта

Возникновение понятия интеллектуальных систем тесно связано с понятием искусственного интеллекта (ИИ) и современной теории управления [1-14]. Исследования в области ИИ стали развиваться наиболее активно с 50-х годов прошлого века в связи с развитием средств вычислительной техники, информатики и кибернетики.

На разных этапах развития ИИ исследователями различных направлений ИИ предлагались свои определения ИИ. Так в математическом энциклопедическом словаре приводилось следующее определение: «Искусственный интеллект (от латинского *intellectus* – познание, понимание, рассудок) это раздел информатики, изучающий методы, способы и приемы моделирования и воспроизведения с помощью ЭВМ разумной деятельности человека, связанной с решением задач». Академик Г.С. Поспелов в книге «Искусственный интеллект – основа новой информационной технологии» писал: «под искусственным интеллектом понимается наука о том, как заставить машину делать то, что умеет делать умный человек». В зарубежной литературе применяют определение термина «искусственный интеллект» как «Artificial Intelligence» (AI). В отличие от философии и психологии, искусственный интеллект в области исследований и разработок технических систем ориентирован не столько на понимание, сколько на построение интеллектуальных систем.

На первоначальном этапе развития были выделены и на данный момент остались актуальными следующие **основные проблемы искусственного интеллекта и направления его развития**:

1. Представление знаний.
2. Решение неформализованных задач.
3. Создание комплексных экспертных систем.
4. Интеллектуальный анализ данных.
5. Общение с ЭВМ на естественном языке.
6. Обучение.
7. Когнитивное моделирование (*cognos* – знание) и моделирование разума.
8. Обработка визуальной информации и робототехника.

Исторически существует 2 направления исследований в области искусственного интеллекта:

1. Моделирование результатов интеллектуальной деятельности или так называемый машинный интеллект.
2. Моделирование биологических систем или так называемый искусственный разум.

Первое направление рассматривает продукт интеллектуальной деятельности человека, изучает его структуру и стремится воспроизвести этот продукт сред-

ствами компьютерной техники. Успехи этого направления тесно связаны с развитием вычислительной техники и программирования.

Второе направление ИИ рассматривает данные о нейрофизиологической деятельности человека. Разработчики стремятся воспроизвести структуры и процессы в нервной системе с помощью технических устройств так, чтобы их поведение хорошо совпадало с поведением человека в заданных ситуациях. Развитие этого направления тесно связано с успехами наук о человеке. В свою очередь работы по искусственному разуму ведутся по следующим направлениям:

1. Моделирование механизмов умственной деятельности, связанное с созданием нейроподобных сетей (нейрокибернетика). В первую очередь изучаются нервные клетки – нейроны и структуры из взаимосвязанных клеток – нейронные сети. Механизмы таких разработок воплощены в нейрокомпьютеры.

2. Моделирование информационных процессов или так называемое эвристическое программирование, при котором моделируются не нейронные клетки, а мыслительные операции, выполнение которых приводит к успешному решению задачи. Успехи теоретического и прикладного характера достигнуты за счет разработки общего решателя задач – GPS (General Problem Solver).

3. Комплексный подход или так называемое эвристическое моделирование.

Необходимо отметить, что современная теория ИИ и созданные на ее основе интеллектуальные системы (ИС) и технологии были подготовлены трудами многих исследователей на протяжении многих лет.

В 40-50-х годах прошлого века началось бурное развитие информационных технологий и автоматизации: были созданы первые ЭВМ, а Норберт Винер создал свои основополагающие работы по кибернетике, и произошло окончательное оформление искусственного интеллекта как научного направления.

В конце 50-х годов XX в. родилась модель лабиринтного поиска. В этом подходе задача представляется как некоторое пространство состояний в форме графа, и в этом графе проводится поиск оптимального пути от входных данных к результирующим.

Начало 60-х годов – это эпоха эвристического программирования. Эвристика – правило, теоретически не обоснованное, но позволяющее сократить количество переборов в пространстве поиска. Эвристическое программирование – разработка стратегии действий на основе известных, заранее заданных, эвристик.

В 1963-1970 гг. к решению задач ИИ стали подключать методы математической логики. На основе метода резолюций француз Альбер Кольмероз в 1973 г. создает язык логического программирования Пролог. Однако большинство реальных задач не сводится к набору аксиом. Человек, решая производственные задачи, не использует классическую логику, и поэтому применение логических моделей при всех своих преимуществах имеют существенные ограничения по классам решаемых задач.

В середине 1970-х годов в США вместо поисков универсальных алгоритма мышления стала развиваться идея моделирования конкретных знаний специа-

листов экспертов. В США появились первые коммерческие системы, основанные на знаниях, или экспертные системы (ЭС). Возник новый подход к решению задач искусственного интеллекта – представление знаний. В это время были созданы две первые классические экспертные системы для медицины (MYCIN) и для химии (DENDRAL).

В конце 70-х годов в Японии было объявлено о начале разработки проекта машин V поколения, основанных на знаниях.

В середине 1980-х годов был разработан в Центре космических исследований NASA язык CLIPS, обеспечивающий поддержку программирования на основе правил объектно-ориентированного и процедурного программирования. Название языка CLIPS – аббревиатура от C Language Integrated Production System.

Начиная с середины 80-х годов, повсеместно происходит коммерциализация искусственного интеллекта. Растут ежегодные капиталовложения, создаются промышленные экспертные системы. Растет интерес к самообучающимся системам, издаются десятки научных журналов, ежегодно собираются международные и национальные конференции по различным направлениям ИИ.

В Советском Союзе работы по искусственному интеллекту начались даже раньше, чем на западе [1-3, 5]. В 1954 г. в МГУ начал свою работу семинар «Автоматы и мышление» под руководством академика А.А. Ляпунова. В этом семинаре принимали участие физиологи, лингвисты, психологи, математики. Принято считать, что именно в это время родился искусственный интеллект в России.

Среди наиболее значимых результатов, полученных отечественными учеными в 60-е годы, отмечают алгоритм «Кора» М.М. Бонгарда, моделировавший деятельность человеческого мозга при распознавании образов.

В 1965-1980 гг. происходит рождение нового направления – ситуационного управления. Основателем этой научной школы стал проф. Д.А. Поспелов, разработавший специальные модели представления ситуаций.

В 1974 г. при Комитете по системному анализу при президиуме АН СССР начал работу Научный совет по проблеме «Искусственный интеллект». В 1988 г. создается Ассоциация Искусственного Интеллекта (АИИ). Ее членами стали более 300 исследователей, а президентом избирается Д. А. Поспелов.

В 1980-1990 гг. проводятся активные исследования в области представления знаний, разрабатываются языки представления знаний, экспертные системы (более 300). В МГУ им. М. В. Ломоносова создается язык РЕФАЛ.

В конце 90-х годов начинают более активно использоваться идеи и методы искусственного интеллекта в теории и практике управления, появляются отдельные разработки на базе ИИ, включаемые в состав систем управления (СУ). Такие разработки стали называть интеллектными компонентами (ИК) систем управления.

В качестве наиболее используемого типа ИК для СУ становятся искусственные нейронные сети. В системах управления применяется несколько типов ИНС: многослойный персептрон, сеть Кохонена (как однослойная самооргани-

зующаяся карта – Kohonen's Self-Organizing Map), сеть Хопфилда, машина Больцмана и другие [5-8].

В качестве второго типа интеллектуальных компонент систем управления рассматриваются эволюционные (генетические) алгоритмы, которые тоже являются инструментом поиска рациональных решений. Эволюционные алгоритмы (ЭА) понимаются как алгоритмы со специальными структурами данных, оперирующие с популяциями индивидов.

Для реализации высокоинтеллектуальных функций систем управления используются интеллектуальные компоненты третьего типа – системы, основанные на «знаниях» (СОЗ, Knowledge-Based Systems). Например, знания могут быть представлены в некотором логическом языке. Обработка знаний с помощью логических средств позволяет получать некоторые предпочтения на множестве допустимых управлений с целью выбора одного из них. В общем случае СОЗ оперируют с более широкой информацией логическими, объектно-ориентированными и другими моделями, основанными на знаниях экспертов. Вместе с тем СОЗ могут использовать и традиционные алгоритмы, базирующиеся на уравнениях динамики.

1.2. Основные направления исследований в области интеллектуальных систем

Рассмотрим краткие характеристики основных направлений исследований в области искусственного интеллекта, описанные в [1-3, 5-11].

Разработка интеллектуальных систем, основанных на знаниях. Это одно из главных направлений ИИ. Основной целью построения таких систем являются выявление, исследование и применение знаний высококвалифицированных экспертов для решения сложных задач, возникающих на практике. При построении систем, основанных на знаниях (СОЗ), используются знания, накопленные экспертами в виде конкретных правил решения тех или иных задач. Это направление преследует цель имитации человеческого искусства анализа неструктурированных и слабоструктурированных проблем. В данной области исследований осуществляется разработка моделей представления, извлечения и структурирования знаний, а также изучаются проблемы создания баз знаний (БЗ), образующих ядро СОЗ. Частным случаем СОЗ являются экспертные системы (ЭС), подробное описание которых можно найти в [1, 5, 9, 17].

Разработка естественно-языковых интерфейсов и машинный перевод. Проблемы компьютерной лингвистики и машинного перевода разрабатываются в ИИ с 1950-х гг. Системы машинного перевода с одного естественного языка на другой обеспечивают быстроту и систематичность доступа к информации, оперативность и единообразие перевода больших потоков. Системы машинного перевода строятся как интеллектуальные системы, поскольку в их основе лежат БЗ в определённой предметной области и сложные модели, обеспечивающие дополнительную трансляцию «исходный язык оригинала – язык смысла – язык перевода». Они базируются на структурно-логическом подходе, включающем последовательный анализ и синтез естественно-языковых сообщений. Кроме

того, в них осуществляется ассоциативный поиск аналогичных фрагментов текста и их переводов в специальных базах данных (БД). Данное направление охватывает также исследования методов и разработку систем, обеспечивающих реализацию процесса общения человека с компьютером на естественном языке (так называемые системы ЕЯ-общения). Кроме того, создаются системы речевого общения в целях повышения скорости ввода информации в ЭВМ, разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии. В таких системах под текстом понимают фонемный текст (как слышится).

Обработка визуальной информации. В этом научном направлении решаются задачи обработки, анализа и синтеза изображений. Задача обработки изображений связана с трансформированием графических образов, результатом которого являются новые изображения. В задаче анализа исходные изображения преобразуются в данные другого типа, например в текстовые описания. При синтезе изображений на вход системы поступает алгоритм построения изображения, а выходными данными являются графические объекты (системы машинной графики).

Обучение и самообучение. Эта актуальная область ИИ включает модели, методы и алгоритмы, ориентированные на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных [19-21]. К данному направлению относятся появившиеся в 90-х годах системы добычи данных (Data Mining) и системы поиска закономерностей в компьютерных базах данных (Knowledge Discovery).

Игры и машинное творчество. Машинное творчество охватывает сочинение компьютерной музыки, стихов, интеллектуальные системы для изобретения новых объектов. Создание интеллектуальных компьютерных игр является одним из самых развитых коммерческих направлений в сфере разработки программного обеспечения. Кроме того, компьютерные игры предоставляют мощный арсенал разнообразных средств, используемых для обучения.

Программное обеспечение систем ИИ. Инструментальные средства для разработки интеллектуальных систем включают специальные языки программирования, ориентированные на обработку символьной информации (LISP, SMALLTALK, РЕФАЛ), языки логического программирования (PROLOG), языки представления знаний (OPS5, KRL, FRL), интегрированные программные среды, содержащие арсенал инструментальных средств для создания систем ИИ (KE, ARTS, GURU, G2), а также оболочки экспертных систем (BUILD, EMYCIN, EXSYS Professional, ЭКСПЕРТ), которые позволяют создавать прикладные ЭС, не прибегая к программированию [7, 15-17].

Новые архитектуры компьютеров. Это направление связано с созданием компьютеров не фон-неймановской архитектуры, ориентированных на обработку символьной информации. Известны удачные промышленные решения параллельных и векторных компьютеров, однако в настоящее время они имеют весьма высокую стоимость, а также недостаточную совместимость с существующими вычислительными средствами.

Интеллектуальные роботы. Создание интеллектуальных роботов составляет конечную цель робототехники. В настоящее время в основном используются программируемые манипуляторы с жёсткой схемой управления, названные роботами первого поколения. Несмотря на очевидные успехи отдельных разработок, эра интеллектуальных автономных роботов пока не наступила. Основными сдерживающими факторами в разработке автономных роботов являются нерешённые проблемы в области интерпретации знаний, машинного зрения, адекватного хранения и обработки трёхмерной визуальной информации.

1.3. Основные признаки и отличия интеллектуальных систем

Для интеллектуальных систем характерны следующие признаки [6-7, 10-11]:

1. Развитые коммуникативные способности: возможность обработки произвольных запросов в диалоге на языке максимально приближенном к естественному (система естественно-языкового интерфейса – СЕЯИ).
2. Направленность на решение слабоструктурированных, плохо формализуемых задач (реализация мягких моделей).
3. Способность работать с неопределёнными и динамичными данными.
4. Способность к развитию системы и извлечению знаний из накопленного опыта конкретных ситуаций.
5. Возможность получения и использования информации, которая явно не хранится, а выводится из имеющихся в базе данных.
6. Система имеет не только модель предметной области, но и модель самой себя, что позволяет ей определять границы своей компетентности.
7. Способность к выводам по аналогии.
8. Способность объяснять свои действия, неудачи пользователя, предупреждать пользователя о некоторых ситуациях, приводящих к нарушению целостности данных.

В отличие от обычных аналитических и статистических моделей, ИС позволяют получить решение трудно формализуемых слабо структурированных задач.

Возможность ИС работать со слабоструктурированными данными подразумевает наличие следующих качеств:

- решать задачи, описанные только в терминах «мягких» моделей, когда зависимости между основными показателями являются не вполне определёнными или даже неизвестными в пределах некоторого класса;
- способность к работе с неопределёнными или динамичными данными, изменяющимися в процессе обработки, позволяет использовать ИС в условиях, когда методы обработки данных могут изменяться и уточняться по мере поступления новых данных;
- способность к развитию системы и извлечению знаний из накопленного опыта конкретных ситуаций увеличивает мобильность и гибкость системы, позволяя ей быстро осваивать новые области применения.

Отличия программирования систем искусственного интеллекта от обычных программных систем представлены в таблице.

Отличия интеллектуальных систем от обычных программных систем

Характеристика	Программирование в интеллектуальных системах	Традиционное программирование
Тип обработки	Символьный	Числовой
Метод	Эвристический поиск	Точный алгоритм
Задание шагов решения	Неявное	Явное
Искомое решение	Удовлетворительное	Оптимальное
Управление и данные	Смешаны	Разделены
Знания	Неточные	Точные
Модификации	Частые	Редкие

Для интеллектуальных информационных систем, ориентированных на генерацию алгоритмов решения задач, характерны следующие признаки:

- развитые коммуникативные способности;
- умение решать сложные плохо формализуемые задачи;
- способность к самообучению.

Коммуникативные способности ИС характеризуют способ взаимодействия (интерфейса) конечного пользователя с системой, в частности, возможность формулирования произвольного запроса в диалоге с ИС на языке, максимально приближенном к естественному.

Сложные плохо формализуемые задачи – это задачи, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, для которой могут быть характерны неопределенность и динамичность исходных данных и знаний.

Способность к самообучению – это возможность автоматического извлечения знаний для решения задач из накопленного опыта конкретных ситуаций.

В различных ИС перечисленные признаки интеллектуальности развиты в неодинаковой степени и редко, когда все четыре признака реализуются одновременно. Условно каждому из признаков интеллектуальности соответствует свой класс ИС:

- системы с интеллектуальным интерфейсом;
- экспертные системы;
- самообучающиеся системы.

Анализ интеллектуальности систем, разработанный профессором Логиновским О.В., основан на следующих базовых интеллектуальных структурах, которые приводятся ниже в соответствии с работой [10]. Такие структуры, как идентификаторы, исполнители, накопители, модификаторы являются базовыми элементами более сложных интеллектуальных систем (экспертных, систем автоматизации управления, в частности, автоматизированных систем проектирования и управления). При этом системы становятся многоуровневыми, иерархическими, на нижнем уровне которых обязательно присутствуют интеллектуальные идентификаторы, исполнители и накопители. В конкретной предметной

области управления (планирования, проектирования) объект управления выделяется в системе идентификаторами и накопителями по принципу обратной связи.

Перечисленные виды интеллектуальных систем:

1. Интеллектуальный идентификатор (ИДЕН)

$$\langle \text{ИДЕН} \rangle = \{ A, p, w, a, Ш \},$$

где A – множество объектов предметной области (опорное множество); p – подмножество A , являющееся классом эквивалентности по какому-либо признаку; w – семантическая достоверность сведения об объекте; a – семантический указатель объекта из опорного множества A ; $Ш$ – шкала понятий для множества A .

Интеллектуальный идентификатор выполняет в системе функции идентификации данных об объектах предметной области по существующим семантическим указателям, решая задачу интеллектуального распознавания объектов.

2. Интеллектуальный исполнитель (ИСП) представляет собой интеллектуальную структуру, позволяющую по данным об объекте a из опорного множества A отыскать конкретный объект и (при необходимости) осуществить операции по его воплощению в предметной области.

3. Интеллектуальный накопитель (НАК) также выполняет функции интеллектуальной системы, состоящие в пополнении данных об объекте от различных источников из множества A , формируя соответствующие базы и банки данных территориального и муниципального управления.

Формульные соотношения интеллектуальных накопителя и исполнителя аналогичны интеллектуальному идентификатору.

4. Интеллектуальный модификатор (МОД) осуществляет функции преобразования данных из одного множества A в данные об объектах другого множества A' со шкалой понятий (перечнем свойств) $Ш'$:

$$\{ A, p, w, a, Ш \} \Rightarrow \{ A', p, w, a', Ш' \}.$$

При совпадении опорных множеств A и A' , данные об объекте $a \in A$ преобразуются модификатором в данные о том же объекте (сингулярные модификаторы). Свойство сингулярности весьма характерно для автоматизированных систем проектирования и управления.

5. Интеллектуальный анализатор (АНАЛ), выполняющий функции анализа и оценки эффективности системы с учетом интеллектуальности и других основополагающих характеристик.

6. Интеллектуальный синтезатор (СИН), осуществляющий синтез новых характеристик, свойств и возможностей системы на основе сформированных баз знаний о предметной области в целом и самой системе в частности с помощью искусственного или естественного интеллекта.

На основе представленного набора интеллектуальных структур можно анализировать или формировать на множестве компонент предметной области топологию интеллектуальности посредством понятия интеллектуальной системы:

$$SI_i = \{ R_i, t_i \},$$

где R_i – множество компонент предметной области для i -той системы; t_i – топология i -той системы, включающую базу топологий в виде набора базовых интеллектуальных структур.

При $t_i = 0$, любая из рассматриваемых систем не является интеллектуальной.

Формализация процедур оценки интеллектуальности систем должна осуществляться совместно с оценкой их эффективности и конкурентоспособности:

$$SE_i = \{ R_i, F_i, C_i, P_i, I_i, E_i \},$$

где F_i – множество компонент функциональной модели i -той системы; C_i – множество компонент структурной модели i -той системы; P_i – множество компонент программного обеспечения i -той системы; I_i – множество компонент информационного обеспечения i -той системы; E_i – множество характеристик эффективности i -той системы.

Все указанные компоненты и характеристики для конкретных систем всегда больше нуля. Более сильная топология соответствует более высокому уровню эффективности и конкурентоспособности системы.

1.4. Основные типы интеллектуальных систем

В данном учебном пособии, как и в [11] мы будем рассматривать следующие классы систем искусственного интеллекта:

1. Системы с интеллектуальной обратной связью и интеллектуальными интерфейсами.
2. Автоматизированные системы распознавания образов.
3. Автоматизированные системы поддержки принятия решений.
4. Экспертные системы.
5. Нейронные сети.
6. Когнитивное моделирование.
7. Генетические алгоритмы и моделирование эволюции.
8. Выявление знаний из опыта (эмпирических фактов) и интеллектуальный анализ данных (Data Mining).

Этими классами системы ИС не исчерпываются, но мы вынуждены ограничиться ими, как основными, в связи с ограниченностью объема учебного пособия.

Системы с интеллектуальным интерфейсом. Применение ИИ для усиления коммуникативных способностей информационных систем привело к появлению систем с интеллектуальным интерфейсом, среди которых можно выделить следующие типы [6-7]:

1. Интеллектуальные базы данных. Они позволяют в отличие от традиционных БД обеспечивать выборку необходимой информации, не присутствующей в явном виде, а выводимой из совокупности хранимых данных.

2. Естественно-языковой интерфейс. Он применяется для доступа к интеллектуальным базам данных, контекстного поиска документальной текстовой информации, голосового ввода команд в системах управления, машинного перевода с иностранных языков. Для реализации ЕЯ-интерфейса необходимо решить проблемы морфологического, синтаксического и семантического анализа, а также задачу синтеза высказываний на естественном языке. При морфологическом анализе осуществляются распознавание и проверка правильности написания слов в словаре.

3. Гипертекстовые системы предназначены для реализации поиска по ключевым словам в базах текстовой информации. Интеллектуальные гипертекстовые системы отличаются возможностью более сложной семантической организации ключевых слов, которая отражает различные смысловые отношения терминов. Таким образом, механизм поиска работает прежде всего с базой знаний ключевых слов, а уже затем непосредственно с текстом. В более широком плане сказанное распространяется и на поиск мультимедийной информации, включающей помимо текстовой и цифровой информации графические, аудио и видео-образы.

4. Системы контекстной помощи можно рассматривать, как частный случай интеллектуальных гипертекстовых и естественно-языковых систем. В отличие от обычных систем помощи, навязывающих пользователю схему поиска требуемой информации, в системах контекстной помощи пользователь описывает проблему (ситуацию), а система с помощью дополнительного диалога ее конкретизирует и сама выполняет поиск относящихся к ситуации рекомендаций. Такие системы относятся к классу систем распространения знаний (Knowledge Publishing) и создаются как приложение к системам документации (например, технической документации по эксплуатации товаров).

5. Системы когнитивной графики позволяют осуществлять интерфейс пользователя с ИС с помощью графических образов, которые генерируются в соответствии с происходящими событиями. Такие системы используются в мониторинге и управлении оперативными процессами. Графические образы в наглядном и интегрированном виде описывают множество параметров изучаемой ситуации. Например, состояние сложного управляемого объекта отображается в виде человеческого лица, на котором каждая черта отвечает за какой-либо параметр, а общее выражение лица дает интегрированную характеристику ситуации.

Экспертные системы. Назначение экспертных систем (ЭС) заключается в решении достаточно трудных для экспертов задач на основе накапливаемой базы знаний, отражающей опыт работы экспертов в рассматриваемой проблемной области. Достоинство применения экспертных систем заключается в возможности принятия решений в уникальных ситуациях, для которых алгоритм заранее не известен и формируется по исходным данным в виде цепочки рассуждений

(правил принятия решений) из базы знаний. Причем решение задач предполагается осуществлять в условиях неполноты, недостоверности, многозначности исходной информации и качественных оценок процессов.

Экспертная система является инструментом, усиливающим интеллектуальные способности эксперта, и может выполнять следующие роли:

- консультанта для неопытных или непрофессиональных пользователей;
- ассистента в связи с необходимостью анализа экспертом различных вариантов принятия решений;
- партнера эксперта по вопросам, относящимся к источникам знаний из смежных областей деятельности.

Область исследования экспертных систем называют «инженерией знаний». ЭС применяются для решения неформализованных проблем, к которым относят задачи, обладающие одной (или несколькими) из следующих характеристик:

- задачи не могут быть представлены в числовой форме;
- неоднозначностью, неточностью, противоречивостью;
- цели нельзя выразить с помощью чётко определённой целевой функции;
- не существует однозначного алгоритмического решения задачи;
- алгоритмическое решение существует, но его нельзя использовать по причине большой размерности пространства решений и ограничений на ресурсы (времени, памяти).

Главное отличие ЭС на рис. 1 и систем искусственного интеллекта от систем обработки данных состоит в том, что в них используется символьный, а не числовой способ представления данных, а в качестве методов обработки информации применяются процедуры логического вывода и эвристического поиска решений.

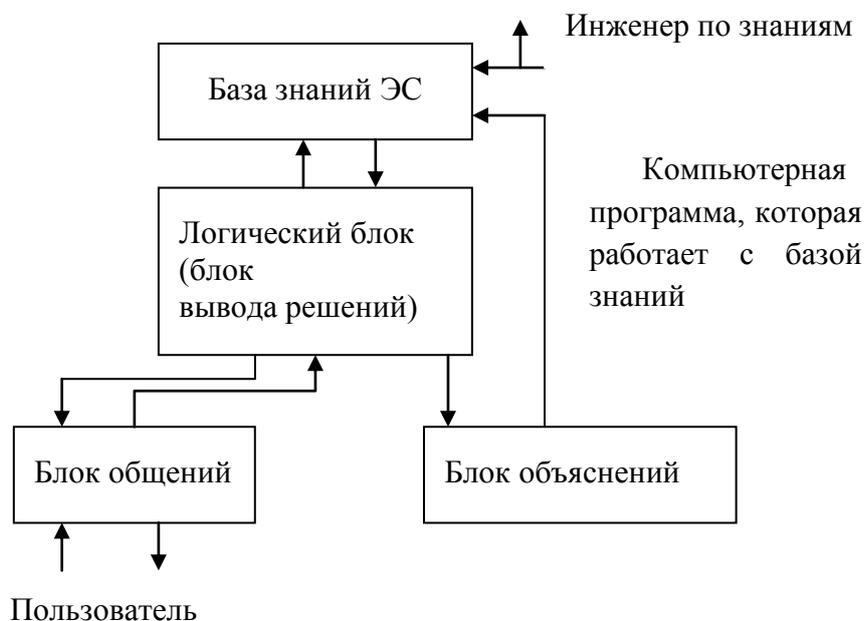


Рис. 1. Общая структура экспертной системы

Отличительным блоком структуры ЭС является логический блок (решатель), в котором вырабатывается экспертное заключение.

Структура классификации экспертных систем представлена на рис.2.

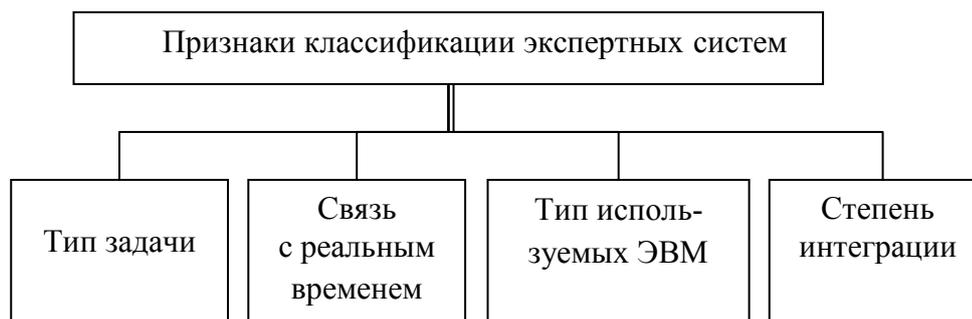


Рис. 2. Классификация экспертных систем

Классификация по решаемой задаче имеет вид:

1. Экспертные системы, обладающие средствами для интеллектуального анализа данных. Для интерпретации данных используются технологии Data Mining.

2. Диагностические экспертные системы. Они применяются при диагнозе технической системы, при диагностике в сфере медицины и в сфере финансов, банковского дела.

3. Экспертные системы мониторинга. Они способны дать объяснение данных в реальном масштабе времени, либо сигнализировать выход параметров за допустимые значения (границы) (например, в подразделениях министерства чрезвычайных ситуаций).

4. Экспертные системы прогнозирования. Системы, которые прогнозируют состояние финансового рынка, курс валют. ЭС прогнозирования являются первыми моделями аналитического плана, т. е. использующими элемент вероятности.

5. Экспертные системы планирования. С помощью таких систем составляется, например, бюджетное планирование, финансирование.

6. Экспертные системы для управления. Это системы, которые готовят варианты возможных решений для управления: например, системы-подсказчики для руководителей.

Классификация ЭС по связи с реальным временем:

1. Статические экспертные системы. К статическим относят такие системы, в которых базы данных и данные не меняются с течением времени (по меньшей мере, во время проведения экспертизы). Примером статической системы может служить ЭС диагностики технического состояния какого-либо устройства.

2. Динамические экспертные системы. Динамические системы применимы для задачи анализа, когда резко меняется ситуация или база знаний. Для работы такой системы должна существовать обратная связь и быстрые машины.

3. Квазидинамические экспертные системы. К квазидинамическим относятся ЭС, в которых если даже за время осуществления экспертизы параметры меняются, то при повторной экспертизе параметры остаются неизменными.

Классификация экспертных систем по типам применяемой архитектуры ЭВМ:

1. Экспертные системы с обычной архитектурой ЭВМ.
2. Экспертные системы со специальными нейрокомпьютерами – многопроцессорными вычислительными системами.

Классификация ЭС по степени интеграции с другими программами:

1. Автономные экспертные системы. К автономным ЭС относят системы, которые не требуют дополнительных программ для обработки данных.
2. Гибридные экспертные системы. К гибридным ЭС относят системы, которые работают с привлечением дополнительных программных средств.

Экспертные системы охватывают самые разные предметные области, среди которых лидируют бизнес, производство, медицина, проектирование и системы управления [1, 5-7, 15, 17, 18].

Нейронные сети. Искусственная нейронная сеть (ИНС) – это кибернетическая модель нервной системы, которая представляет собой совокупность большого числа сравнительно простых элементов – нейронов, топология соединения которых зависит от типа сети. Чтобы создать нейронную сеть для решения какой-либо конкретной задачи, следует выбрать способ соединения нейронов друг с другом и подобрать значения параметров межнейронных соединений.

Нейронные сети – обобщённое название группы математических алгоритмов, обладающих способностью обучаться на примерах, «узнавая» впоследствии черты встреченных образцов и ситуаций. Благодаря этой способности нейронные сети используются при решении задач обработки сигналов и изображений, распознавания образов, а также для прогнозирования [7].

ИНС представляет собой совокупность простых вычислительных элементов – искусственных нейронов, каждый из которых обладает определённым количеством входов (дендритов) и единственным выходом (аксоном), разветвления которого подходят к синапсам, связывающим его с другими нейронами. На входы нейрона поступает информация извне или от других нейронов. Каждый нейрон характеризуется функцией преобразования входных сигналов в выходной (функция возбуждения нейрона). Нейроны в сети могут иметь одинаковые или разные функции возбуждения. Сигналы, поступающие на вход нейрона, неравнозначны в том смысле, что информация из одного источника может быть более важной, чем из другого. Приоритеты входов задаются с помощью вектора весовых коэффициентов, моделирующих синаптическую силу биологических нейронов.

Модель искусственного нейрона представляет собой дискретно-непрерывный преобразователь информации. Информация, поступающая на вход нейрона, суммируется с учётом весовых коэффициентов w_i , сигналов x_i , $i = 1, \dots, n$, где n – размерность пространства входных сигналов. Потенциал нейрона определяется по формуле:

$$P = \sum_{i=0}^n x_i w_i.$$

Взвешенная сумма поступивших сигналов (потенциал) преобразуется с помощью передаточной функции $f(P)$ в выходной сигнал нейрона Y , который передается другим нейронам сети, т. е. $Y = f(P)$. Вид передаточной (активационной) функции является важнейшей характеристикой нейрона. В общем случае эта функция может быть ступенчатой (пороговой), линейной или нелинейной). Тип функции переноса выбирается с учётом конкретной задачи, решаемой с применением нейронных сетей. Например, в задачах аппроксимации и классификации предпочтение отдают логистической (сигмоидальной) кривой.

Нейронная сеть представляет собой совокупность искусственных нейронов, организованных слоями. При этом выходы нейронов одного слоя соединяются с входами нейронов другого.

В зависимости от топологии соединений нейронов ИНС подразделяются на одноуровневые и многоуровневые, с обратными связями и без них. Связи между слоями могут иметь различную структуру.

В однолинейных сетях каждый нейрон (узел) нижнего слоя связан с одним нейроном верхнего слоя. Если каждый нейрон нижнего слоя соединён с несколькими нейронами следующего слоя, то получается пирамидальная сеть. Воронкообразная схема соединений предполагает связь каждого узла верхнего слоя со всеми узлами нижнего уровня. Существуют также древовидные и рекуррентные сети, содержащие обратные связи с произвольной структурой межнейронных соединений.

Чтобы построить ИНС для решения конкретной задачи, нужно выбрать тип соединения нейронов, определить вид передаточных функций элементов и подобрать весовые коэффициенты межнейронных связей.

При всём многообразии возможных конфигураций ИНС на практике получили распространение лишь некоторые из них. Классические модели нейронных сетей:

- многослойные сети;
- сети Хопфилда;
- сети Хемминга;
- сети обратного распространения;
- сети встречного распространения;
- РБФ-сети;
- сети Кохонена и др.

С помощью нейронных сетей, которые относят к методам эволюционного моделирования, решается широкий класс задач: классификация образов, кластеризация, аппроксимация, прогноз данных, оптимизация, ассоциативная память, управление динамическими объектами. Причем в силу всего вышесказанного, нейронные сети в сравнении с методами математической статистики справляются с перечисленными задачами тем успешнее, чем хуже формализуема задача.

Нейросетевые технологии нашли широкое применение в таких направлениях, как распознавание печатного текста, контроль качества продукции на производстве, идентификация событий в ускорителях частиц, разведка нефти,

борьба с наркотиками, медицинские и военные приложения, управление и оптимизация, финансовый анализ, прогнозирование и др.

В сфере экономики нейросетевые технологии могут использоваться для классификации и анализа временных рядов путём аппроксимации сложных нелинейных функций.

Экспериментально установлено, что модели нейронных сетей обеспечивают большую точность при выявлении нелинейных закономерностей на фондовом рынке по сравнению с регрессионными моделями [7].

Когнитивное моделирование – это способ анализа, обеспечивающий определение силы и направления влияния факторов на перевод объекта управления в целевое состояние с учетом сходства и различия в влиянии различных факторов на объект управления.

Ведущей научной организацией России, занимающейся разработкой и применением технологии когнитивного анализа, является Институт проблем управления РАН. На их научных трудах в области когнитивного анализа и основывается данная лекция [11].

В основе технологии когнитивного анализа и моделирования (рис. 3) лежит когнитивная (познавательная-целевая) структуризация знаний об объекте и внешней для него среды.

Когнитивная структуризация предметной области – это выявление будущих целевых и нежелательных состояний объекта управления и наиболее существенных (базисных) факторов управления и внешней среды, влияющих на переход объекта в эти состояния, а также установление на качественном уровне причинно-следственных связей между ними, с учетом взаимовлияния факторов друг на друга.

Результаты когнитивной структуризации отображаются с помощью когнитивной карты (модели).

Отбор базисных факторов проводится путем применения *PEST-анализа*, выделяющего четыре основные группы факторов (аспекта), определяющих поведение исследуемого объекта:

- **Policy** – политика;
- **Economy** – экономика;
- **Society** – общество (социокультурный аспект);
- **Technology** – технология.

PEST-анализ можно рассматривать как вариант системного анализа, так как факторы, относящиеся к перечисленным четырем аспектам, в общем случае тесно взаимосвязаны и характеризуют различные иерархические уровни общества, как системы.

В этой системе есть детерминирующие связи, направленные с нижних уровней иерархии системы к верхним (наука и технология влияет на экономику, экономика влияет на политику), а также обратные и межуровневые связи. Изменение любого из факторов через эту систему связей может влиять на все остальные.

Эти изменения могут представлять угрозу развитию объекта, или, наоборот, предоставлять новые возможности для его успешного развития.

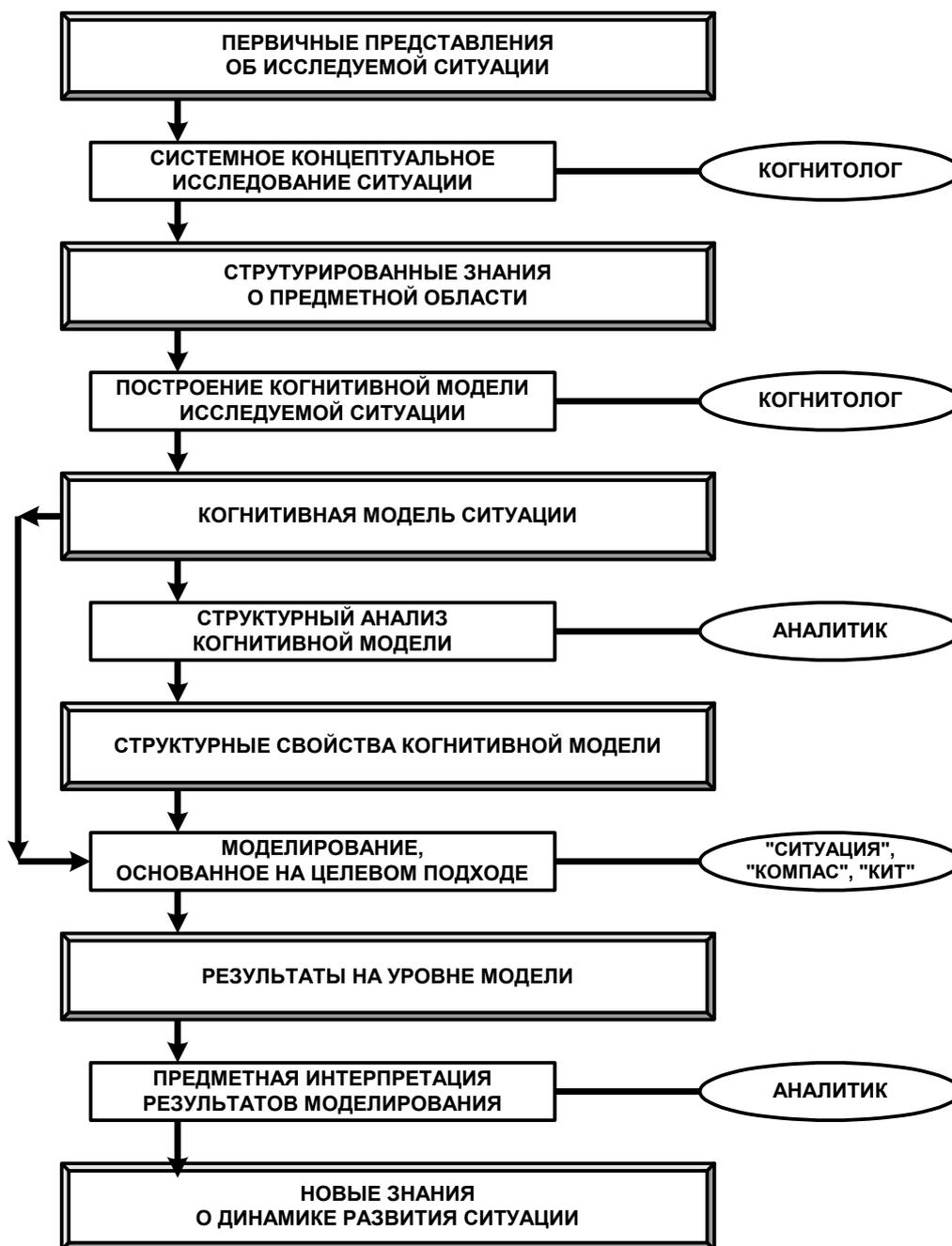


Рис. 3. Технология когнитивного анализа и моделирования

Следующий шаг – ситуационный анализ проблем, так называемый *SWOT-анализ*:

- **Strengths** – сильные стороны;
- **Weaknesses** – недостатки, слабые стороны;
- **Opportunities** – возможности;
- **Threats** – угрозы.

SWOT-анализ включает анализ сильных и слабых сторон развития исследуемого объекта в их взаимодействии с угрозами и возможностями и позволяет определить актуальные проблемные области, узкие места, шансы и опасности, с учетом факторов внешней среды.

Технология когнитивного анализа и моделирования поддерживается программными комплексами «Ситуация», «Компас», «КИТ» (см. рис. 3), созданными в ИПУ РАН, которые позволяют в сложных и неопределенных ситуациях быстро, комплексно и системно охарактеризовать и обосновать сложившуюся ситуацию и на качественном уровне предложить пути решения проблемы в этой ситуации с учетом факторов внешней среды.

Применение когнитивных технологий открывает новые возможности прогнозирования и управления в различных областях:

– в экономической сфере – позволяет в сжатые сроки разработать и обосновать стратегию экономического развития предприятия, банка, региона или даже целого государства с учетом влияния изменений во внешней среде;

– в сфере финансов и фондового рынка – учесть ожидания участников рынка;

– в военной области и области информационной безопасности – противостоять стратегическому информационному оружию, заблаговременно распознавая конфликтные структуры и вырабатывая адекватные меры реагирования на угрозы.

Когнитивные технологии автоматизируют часть функций процессов познания, поэтому они с успехом могут применяться во всех областях, в которых востребовано само познание. Вот лишь некоторые из этих областей:

1. Модели и методы интеллектуальных информационных технологий и систем для создания геополитических, национальных и региональных стратегий социально-экономического развития.

2. Модели выживания «мягких» систем в изменяющихся средах при дефиците ресурсов.

3. Ситуационный анализ и управление развитием событий в кризисных средах и ситуациях.

4. Информационный мониторинг социально-политических, социально-экономических и военно-политических ситуаций.

5. Разработка принципов и методологии проведения компьютерного анализа проблемных ситуаций.

6. Выработка аналитических сценариев развития проблемных ситуаций и управления ими.

7. Подготовка рекомендаций по решению первоочередных стратегических проблем на основе компьютерной системы анализа проблемных ситуаций.

8. Мониторинг проблем в социально-экономическом развитии корпорации, региона, города, государства.

9. Технология когнитивного моделирования целенаправленного развития региона РФ.

10. Анализ развития региона и мониторинг проблемных ситуаций при целенаправленном развитии региона.

Интеллектуальный анализ данных. Данная лекция основана на работах [8, 19-21]. К числу интеллектуальных технологий анализа данных относят технологии *Data Mining* (разведка, добыча данных), которые реализуют:

- поиск ранее неизвестных функциональных и логических закономерностей накопленной информации (интеллектуальная деятельность);
- построение моделей и правил с целью объяснения этих найденных закономерностей или найденных аномалий;
- прогнозирование развития изучаемых процессов.

Специфика современных требований к переработке данных:

- данные имеют практически неограниченный объем;
- данные являются разнородными (количественными, качественными, тестовыми и т.д.);
- результаты должны быть конкретны и понятны;
- инструменты для обработки «сырых», т. е. непроверенных, невалидированных данных должны быть просты в использовании.

Процесс обнаружения в «сырых» данных включает в себя поиск:

- ранее неизвестных знаний;
- нетривиальных выводов;
- практически полезных выводов;
- доступных для интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Место технологий *Data Mining* среди других технологий обработки данных по уровню знаний, извлекаемых из данных, приведено на рис 4.



Рис. 4. Схема уровней знаний, извлекаемых из данных

Data Mining применяется в следующих отраслях:

- торговля, в том числе и розничная (анализ покупательской корзины, анализ временных шаблонов, создание прогнозируемых моделей);
- страхование;
- банковское дело (выявление мошенников с кредитными карточками, сегментация клиентов (разбиение на группы – каким клиентам какие услуги предлагать), прогнозирование изменения клиентуры);
- в сфере коммуникаций (здесь ставятся следующие задачи: анализ записей по характеристикам вызовов – какой круг клиентов, какими видами услуг пользуется; выявление лояльности клиентов);
- химия;
- промышленность;
- медицина и т. д.

Успех применения систем Data Mining основан на том, что эти технологии обеспечивают исследование эмпирических данных и выявление в них скрытых закономерностей различных видов, т. е. обеспечивают решение следующих задач:

1. Классификация – задача выявления признаков однотипных объектов для того, чтобы отнести новый объект к тому или иному классу.
2. Кластеризация – развитие идеи классификации на более сложный случай, когда сами классы объектов не predetermined, но стоит задача определить однородные группы данных.
3. Выявление ассоциаций – установление правил, указывающих на то, что при наступлении одного события с определенной вероятностью наступает другое.
4. Выявление последовательностей – установление связи между событиями, наступающими не одновременно, а с некоторым временным шагом.
5. Задача прогнозирования – оценка будущих значений, определяемых показателей на основе их текущих и ретроспективных данных. Это наиболее сложная из перечисленных задач.

Процесс информационной деятельности может быть представлен рядом временных стадий, которые изображены на рис.5.

Свободный поиск – это процесс исследования накопленных данных на предмет поиска скрытых закономерностей без предварительного выявления гипотез относительно этих закономерностей (если гипотеза скрытая, то поиск свободный).

В процессе свободного поиска используется математический аппарат условной логики, с помощью правил которой могут быть описаны группы примеров в задачах классификации и кластеризации.

На основе использования правил ассоциативной логики можно решать задачи выявления ассоциаций и последовательностей.

Выявление трендов (наличие какой-то устойчивой закономерности в том объеме данных, которые используются) и колебаний является первым этапом программирования.

Стадии					
Свободный поиск			Прогностическое моделирование		Анализ исключений
Выявление закономерностей условной логики	Выявление закономерностей ассоциативной логики	Выявление трендов и колебаний	Предсказание неизвестных значений	Предсказание развития процессов	Выявление и объяснение отклонений
Действия					

Рис. 5. Стадии информационной деятельности

Вторая стадия информационной деятельности – прогностическое моделирование. Осуществление действий на этой стадии становится возможным после выполнения действий на стадии свободного поиска.

Третья стадия – анализ исключения из правил. Данная стадия занимается тем, чем обусловлено выпадение какого-то параметра от ранее установленной закономерности, выявляются отклонения и даются им объяснения.

Математической основой систем Data Mining являются следующие методы поиска различных закономерностей в данных: деревья решений; алгоритмы кластеризации; регрессионный анализ; нейронные сети; временные ряды и др. Указанные методы и программные продукты Data Mining подробнее описаны в главе 3.

Генетические Алгоритмы (ГА) – это адаптивные методы функциональной оптимизации, основанные на компьютерном имитационном моделировании биологической эволюции. Основные принципы ГА были сформулированы Голландом в 1975 году и хорошо описаны во многих работах и на ряде сайтов в Internet [7-8].

В генетических алгоритмах используются специфические термины, взятые из генетики, которые трактуются следующим образом. Исходные логические закономерности в базе данных именуют хромосомами, а весь набор закономерностей называют популяцией хромосом. Популяция обрабатывается с помощью процедур репродукции, изменчивости (мутаций), генетической композиции. В ходе работы процедур на каждой стадии эволюции получают популяции со все более совершенными индивидуумами.

Генетические алгоритмы сконструированы таким образом, что при генерации каждой новой популяции используются фрагменты исходных решений, к которым добавляются новые элементы, обеспечивающие улучшение решений относительно сформулированного критерия отбора. Другими словами, генети-

ческие алгоритмы используют информацию, накопленную в процессе эволюции.

В генетических алгоритмах и эволюционном программировании используют два основных механизма воспроизводства хромосом:

- воспроизводство без мутаций, соответствующее митозу, результатом которого являются потомки – копии родителей;
- воспроизводство потомков, имеющих большие отличия от родителей. Этот механизм соответствует половому размножению.

В генетических алгоритмах в основном используется механизм родительского воспроизводства с рекомбинацией и мутацией, а в эволюционном программировании – механизм на основе мутации без рекомбинации.

Таким образом, по сути дела каждый конкретный генетический алгоритм представляют имитационную модель некоторой определенной теории биологической эволюции или ее варианта. Материальное воплощение сконструированных таким образом систем до сих пор была невозможна без участия человека. Однако интенсивно ведутся работы, результатом которых является уменьшение зависимости машинной эволюции от человека. Эти работы ведутся по двум основным направлениям:

1. Естественный отбор, моделируемый ГА, переносится из виртуального мира в реальный, например, проводятся эксперименты по реальным битвам роботов на выживание.

2. Интеллектуальные системы, основанные на ГА, конструируют роботов, которые в принципе могут быть изготовлены на автоматизированных заводах без участия человека.

Генетические алгоритмы используют в теории нечётких систем для настройки параметров функций принадлежности. Интеграция чётких и нечётких нейронных сетей и генетических алгоритмов обеспечивает реализацию оптимизационной задачи. Средства fuzzy–neuro–genetic используются в интеллектуальных системах и содержат следующие процедуры:

- преобразование входных примеров в нечёткое представление;
- извлечение знаний, представленных в виде продукций ЕСЛИ–ТО из нечёткой обучающей выборки с помощью нейронной сети;
- оптимизацию структуры продукционных правил с помощью генетического алгоритма.

Активно развивается направление, ориентированное на использование генетических алгоритмов для обучения нейронных сетей и корректировки структуры уже обученной сети. В отличие от метода обратного распространения ошибки генетические алгоритмы мало чувствительны к архитектуре сети. Генетические алгоритмы чаще всего применяются для улучшения характеристик ИНС, уже созданных и обученных с применением других методов

На основе генетических алгоритмов предложены классифицирующие системы, которые можно использовать для целей управления. Классифицирующая система состоит из трёх вложенных друг в друга подсистем: классификатора, системы обучения и генетического алгоритма. В классификатор поступают

внешние сообщения и положительные оценки (поощрения) его действий. Классификатор содержит правила вида «ЕСЛИ<условие>, ТО<сообщение>», с помощью которых формируются выходные сообщения. Обучающая система выполняет оценку используемых правил. Генетический алгоритм предназначен для случайно направленной модификации правил.

В настоящее время активно развиваются методы, основанные на объединении технологий инженерии знаний и генетических алгоритмов. В области ГА разрабатываются операторы, ориентированные на обработку знаний.

Генетические алгоритмы нашли широкое практическое применение в менеджменте и управлении для решения задач поиска оптимальных решений, формирования моделей и прогнозирования значений различных показателей. Они осуществляют поиск лучших решений на основе заданной целевой функции. Значение целевой функции для многих задач весьма непросто вычислить, поэтому в ряде случаев при исследовании плохо обусловленных проблем с этой целью применяются нейронные сети, позволяющие найти решение при отсутствии явной модели. Кроме того, для вычисления целевых функций в условиях неопределённости применяются статистические методы и методы логического вывода в чёткой или нечёткой среде.

Формирование системы прогнозирующих правил. Генетические алгоритмы могут использоваться для нахождения оптимального набора правил, позволяющих прогнозировать страховые риски с учётом ряда определяющих его факторов. Для решения этой задачи необходимо иметь базу данных, содержащую фактические значения переменных, влияющих на страховой риск.

Выводы

Современное состояние разработок в области интеллектуальных систем в России можно охарактеризовать как стадию все возрастающего интереса среди широких слоев специалистов – менеджеров, инженеров, программистов и других. Наибольшие трудности в разработке интеллектуальных систем, основанных на знаниях, вызывает не столько процесс машинной реализации систем, а этап анализа знаний и проектирования базы знаний.

Создание интеллектуальных систем должно включать три этапа:

- создание материальной системы поддержки (эта проблема в основном решена, так как ИС могут создаваться даже на базе современных персональных компьютеров);
- создание системы потенциального искусственного интеллекта, то есть программной оболочки, инструментальной системы (таких систем в настоящее время существует пока еще очень мало);
- обучение и самообучение системы искусственного интеллекта и преобразование ее в реальную ИС.

Интеллектуальные системы применяются для решения сложных задач, связанных с использованием слабо формализованных знаний специалистов-практиков, а также с логической обработкой информации. Например, поддерж-

ка принятия решения в сложных ситуациях, анализ визуальной информации, управление в социально-экономической сфере.

Контрольные вопросы

1. Перечислите основные проблемы искусственного интеллекта и направления его развития.
2. Назовите основные направления исследований в области искусственного интеллекта.
3. Какова история исследований в области искусственного интеллекта в нашей стране и за рубежом?
4. Перечислите признаки интеллектуальных информационных систем.
5. Сформулируйте характеристики базовых интеллектуальных структур для анализа интеллектуальности систем.
6. Дайте определение понятия «интеллектуальная система», ее место в классификации информационных систем.
7. Сформулируйте основные отличия интеллектуальных систем от обычных программных систем.
8. Приведите классификацию интеллектуальных систем, цели и пути их создания.
9. Перечислите основные типы систем с интеллектуальным интерфейсом и дайте им краткую характеристику.
10. Перечислите основные типы экспертных систем и дайте им краткую характеристику.
11. Перечислите и охарактеризуйте основные компоненты экспертных систем.
12. Каков вид передаточной (активационной) функции нейрона?
13. Назовите классические модели нейронных сетей.
14. Назовите и дайте краткую характеристику базовым архитектурам нейронных сетей.
15. Охарактеризуйте место технологий интеллектуального анализа данных (Data Mining) среди других технологий обработки данных.
16. Сформулируйте задачи и стадии интеллектуального анализа данных и типы определяемых закономерностей.
17. Перечислите основные направления эволюционного моделирования и приведите основные факторы, определяющие неизбежность эволюции.
18. Какие алгоритмы называют генетическими? Сформулируйте основные особенности генетических алгоритмов.
19. Как проводится отбор базисных факторов управления и внешней среды, влияющих на переход объекта в будущие целевые и нежелательные состояния при когнитивной структуризации предметной области?
20. Назовите области применения когнитивных технологий для решения задач прогнозирования и управления.

Глава 2. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

2.1. Представление знаний, рассуждений и задач

Систему знаний (СЗ) можно определить как некую математическую модель определенной области прикладного неформализованного знания. Система понятий и отношений этой математической модели должны отображать систему понятий и отношений прикладного знания, а зависимости, существующие в этой модели, аппроксимируют соответствующие зависимости прикладного знания [5, 8, 12, 25].

Разработанные модели должны быть зафиксированы в памяти ЭВМ и использоваться для решения с помощью прикладных программ.

Создание интеллектуальных систем, основанных на знаниях, связано с решением следующих взаимосвязанных проблем:

1. Проблема формализации знаний. Эта задача решается с помощью специалистов-прикладников и математиков. Результатом ее будет разработка концептуальной схемы модели.

2. Проблема представления знаний. Ее решение состоит в разработке формального аппарата для фиксации знаний в памяти ЭВМ.

3. Проблема использования знаний. Суть ее заключается в необходимости разработки теории вычислений и преобразований в построенных ранее моделях.

4. Проблема создания баз знаний и систем управления ими. Эта задача для системных программистов по разработке средств программной поддержки моделей знаний.

Знания о предметной области и способах решения в ней задач весьма разнообразны. Возможны различные классификации этих знаний, которые подробно описаны в литературе по управлению знаниями, в частности в [12, 25].

Требования к системам знаний. Выделяют следующие особенности, которыми должна обладать система знаний:

– Терпимость к противоречиям, которая вытекает из открытости внешнего мира и соответственно неполноты знаний о нем.

– Обеспечение вывода, т.е. система знаний должна быть способна к логическому выводу как из уже имеющейся информации, так и из вновь поступающих сведений. Различают два типа вывода:

– свободный (вывод осуществляется при поступлении новой информации в СЗ);

– направленный (вывод запускается при поступлении в СЗ конкретного запроса).

– Критичность к новой информации. Это способность проверить достоверность новой информации и согласовать её с имеющимися знаниями.

– Дробность системы знаний – знания в системе должны быть разбиты на фрагменты, каждый из которых может быть эффективно использован.

– Обучаемость и способность к переструктурированию знаний, т. е. в ходе функционирования должна обеспечиваться перестройка структуры СЗ, повышающая ее эффективность.

Особенности знаний для их представления в ЭВМ. Для представления СЗ в компьютере она должна обладать следующими особенностями:

1. Внутренняя интерпретируемость – каждая информационная единица должна иметь уникальное имя, по которой бы информационная система находила ее, а также отвечала бы на запросы, в которой это имя упомянуто.

2. Структурированность знаний, т. е. информационные единицы должны обладать определенной структурой, это означает, что должны быть возможности для установления отношений типа: часть – целое или род – вид, или элемент – класс.

3. Связность – в информационной базе между информационными единицами должна быть возможность установления связей различного типа, характеризующих отношения между ними. Семантика отношений может носить декларативный или процедурный характер.

Различают отношения структуризации, функциональные отношения типа аргумент – функция, казуальные отношения (причинно – следственные связи) и семантические отношения (объединяют и представляют все ранее перечисленные связи).

Перечисленные особенности позволяют создавать общую модель знаний, которую называют «семантическая сеть». Она представляет модель знаний, в вершинах которой находятся информационные единицы, а дуги характеризуют виды связей между информационными единицами. Если связи иерархичны, то они определяют отношения структуризации, а неиерархические связи будут определять отношения иных типов.

4. Семантическая метрика – отношение, которое характеризует ситуационную близость информационных единиц, т.е. силу ассоциативных связей между ними.

5. Активность. Появление новых данных должны стать источником активности интеллектуальной системы, т.е. выполнение программ в ИС должно инициироваться текущим состоянием информационной базы.

Перечисленные пять особенностей информационных единиц определяют ту грань, за которой данные превращаются в знания, базы данных перерастают в базы знаний (БЗ), а совокупность средств, обеспечивающих работу со знаниями, образуют систему управления базой знаний (СУБЗ).

Таким образом, база знаний (англ. Knowledge Base, KB) – это особого рода база данных, разработанная для управления знаниями (метаданными), то есть сбором, хранением, поиском и выдачей знаний. Раздел искусственного интеллекта, изучающий базы знаний и методы работы со знаниями, называется инженерией знаний. Под базами знаний понимают совокупность фактов и правил вывода, допускающих логический вывод и осмысленную обработку информации.

База знаний – важный компонент интеллектуальной системы. Наиболее известный класс таких программ – экспертные системы. Они предназначены для построения способа решения специализированных проблем, основываясь на записях БЗ и на пользовательском описании ситуации.

Создание и использование систем искусственного интеллекта потребует огромных баз знаний.

2.2. Модели представления знаний

В интеллектуальных системах в соответствии с [2, 5, 8] наиболее распространенными являются следующие способы представления знаний:

1. Логические модели.
2. Сетевые модели.
3. Фреймовые модели.
4. Продукционные модели.
5. Другие методы представления знаний.

Логические модели. В основе модели лежит формальная система, которая на языке теории множеств описывается следующей четверкой множеств:

$$M = \langle T, P, A, B \rangle,$$

где T – это множество базовых элементов различной природы, представляющих тезаурус, например: термины из какого либо словаря, набор деталей из конструкции; P – это множество синтаксических правил, с помощью которых из элементов множества T образуют синтаксически правильные совокупности; A – элементы этого множества образуют аксиомы, определенные на множестве P ; B – множество правил вывода, которые применяются к элементам множества A .

Для множества T существует некоторый способ $\Pi(T)$ определения принадлежности к этому множеству, а для определения принадлежности к множеству A вводится процедура $\Pi(A)$.

Во множество A входят информационные единицы, которые введены в базу извне, а с помощью правил вывода из них получают новые производные знания. Таким образом, описанная система представляет собой генератор порождения новых знаний в данной предметной области.

Логические модели представления знаний реализуются с помощью логики предикатов. Предикатом называют функцию, принимающую только два значения: истина и ложь и предназначенную для выражения свойств объектов или связей между ними. Кроме того, могут использоваться логические знаки конъюнкции (\vee), импликации (\wedge), логического следования (\Rightarrow), кванторы \forall, \exists и другие.

Рассмотрим в качестве примера знание: «Когда температура в печи достигает 120° и прошло менее 30 мин с момента включения печи, то давление не превосходить критического. Если с момента включения прошло более 30 мин, то необходимо включить вентиль N2».

В этой записи первая строка представляет собой описание декларативного знания, а вторая строка описывает процедурные знания.

С помощью логики предикатов, указанное утверждение выглядит следующим образом:

$$P(p = 120) T(t < 30) \rightarrow (D < D_{кр.}); \\ P(p = 120)T(t > 30) \Rightarrow F(N2),$$

где $P(p = 120)$ создает предикат, который становится истинным, когда температура достигается 120° ; $T(t < 30)$ – предикат, остающийся истинным в течение 30 мин с начала процесса; $T(t > 30)$ – предикат, становящийся истинным по истечении 30 мин; $(D < D_{кр.})$ – это утверждение о том, что давление ниже критического; $F(N2)$ – команда открыть вентиль $N2$.

Разработанные системы логического программирования Пролог, в ответах на простейшие запросы к базам знаний, выдают значения «истина» и «ложь» в зависимости от наличия соответствующих фактов [15, 16]. Преимущества логической модели представления знаний заключается в возможности непосредственно запрограммировать механизм вывода правильных высказываний.

Подобные языки логического типа использовались на ранних стадиях развития интеллектуальных систем, однако впоследствии большее внимание стали уделять другим моделям, что связано с такими недостатками логической модели, как отсутствие наглядности и удобочитаемости, а также громоздкость записи, которая приводит к трудности нахождения ошибок. Еще один недостаток состоит в том, что большинство интеллектуальных задач характеризуется недостаточной полнотой, неточностью и некорректностью. Это важно, например, в медицине и биологических науках, так как они слабо формализованные. Поэтому в данных областях науки чаще используют методы аналогий и ассоциаций.

Сетевые модели. В основе этой модели лежит конструкция, которая называется семантической сетью, которая может быть описана на языке теории множеств:

$$H = \langle I, C1, C2, \dots, Cn, G \rangle,$$

где I – множество информационных единиц; $C1, C2, \dots, Cn$ – множество типов связей между информационными единицами; G – множество отображений между информационными единицами.

В зависимости от типов связей между информационными единицами, которые используются в данной модели, различают следующие типы сетей:

- Классифицирующие сети – применяют отношение структуризации.
- Функциональные сети – функциональные отношения между информационными единицами. Примером могут выступать вычислительные модели.
- Сценарии.

В сценариях часто используются казуальные отношения (причинно-следственные отношения) между информационными единицами. Кроме того, могут встречаться отношения следующих типов «средство – результат», «орудие – действие» и т. п.

Так семантическая сеть, представляющая знания об автомобиле гр. Васильева, показана на рис. 6.

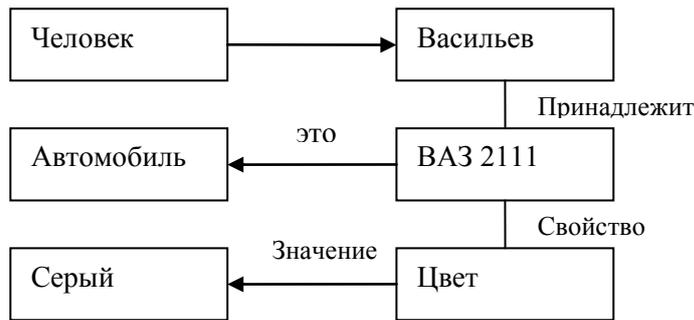


Рис. 6. Пример семантической сети

Преимущества модели представления знаний на основе семантических сетей: с помощью этой модели могут быть представлены родовидовые отношения; простота и наглядность описания предметной области.

Недостатки модели представления знаний на основе семантических сетей: снижение скорости работы механизма вывода, а также понижение быстродействия.

Фреймовые модели. Фреймы часто используют как структуру для представления стереотипных ситуаций. Фрейм – единица представления знаний, детали которой могут изменяться в соответствии с текущей ситуацией.

Структура фрейма такова, что он состоит из характеристик описанных ситуаций и их значений, которые называются соответственно слотом и заполнителем слота.

Имя фрейма:

Имя слота 1 (Значение слота 1);

Имя слота 2 (Значение слота 2);

.....

Имя слота k (Значение слота k).

Эта структура, пока она не заполнена какими-то значениями, называется протофреймом. Значением слота может быть число, математическое выражение, текст на естественном языке, программа для ЭВМ, правила вывода, ссылки на другие слоты данного фрейма или других фреймов.

При заполнении фрейма, имени слота и значения слотов из протофрейма получают фрейм-экземпляр (экзофрейм).

Пример:

Список работников: Фамилия (значение слота 1);

Год рождения (значение слота 2);

.....

Стаж работы (значение слота k).

Список работников: Фамилия (Попов – Сидоров – Иванов);

Год рождения (1965 – 1946 – 1925);

.....

Стаж работы (5 – 20 – 30).

Преимущество фреймовой модели заключается в том, что значения слотов представляются в такой системе в единственном экземпляре, поскольку включается только в один фрейм. Таким образом, обеспечивается экономное размещение баз знаний в памяти компьютера.

Недостатком фреймовой модели является снижение скорости работы механизма вывода, т. е. системы должны «просканировать» всю структуру на предмет ненужной информации.

Продукционные модели. Психологические исследования процессов принятия человеком решений показали, что, рассуждая и принимая решения человек, использует правила, аналогичные продукциям. Продукционная модель, основанная на правилах, позволяет представить знания в виде предложений типа «если (условие), то (действие)».

В общем случае продукцию можно представить в виде следующего выражения:

$$(i); Q; P; A \Rightarrow B; N,$$

где (i) – имя продукции, с помощью которой данная продукция выделяется из всего множества (порядковый номер, некоторая совокупность слов или цифр).

Элемент Q характеризует сферу применения продукции.

Основным элементом продукции является ее ядро $A \Rightarrow B$. Оно может быть прочитано: если выполняется A , то следует B . Ядро продукции описывает преобразования, которые представляют суть продукционного правила.

Элемент P есть условие применимости ядра продукции. Обычно P представляет собой логическое выражение типа предиката. Если P принимает значение «истина», то ядро продукции активируется, если P ложно, то ядро продукции не может быть использовано.

Элемент N описывает постусловие продукции. Оно активируется, когда ядро продукции реализовано. N – это действия и условия процедуры, которые могут быть выполнены после P .

Таким образом, продукционная модель хорошо применима для представления процедурных знаний.

Имя продукции i – интерпретация результатов психологического тестирования.

Предусловие Q – использовать в первую очередь.

Условие P – шкала лжи $L < 70$.

Ядро $A \Rightarrow B$ – если (шкала ошибок – шкала коррекции) $< - 11$, то сообщение: «Результаты недостоверны».

Постусловие N – закончить интерпретацию результатов.

Все ядра можно разделить на 2 большие группы:

1. Детерминированные ($A \Rightarrow B$ наступит с вероятностью 1).
2. Недетерминированные (если A выполняется, то возможно B).

Возможны различные оценки реализации ядра:

1. Вероятностная (если выполняется A , то с вероятностью P реализуется B).

2. Лингвистическая (малая, меньшая). Если выполняется A , то с большой долей уверенности наступит B .

Детерминированные продукции могут быть:

1. Однозначными (если выполняется A , то наступит B).

2. Альтернативные (если A выполняется, то чаще наступит $B1$, реже $B2$).

Продукции могут быть прогнозирующего типа (если наступит A , то с вероятностью P можно ожидать B).

Продукционной называется система знаний, которая использует понятие продукции в качестве основного элемента.

В общем случае продукционная система знаний (см. рис. 1) включает следующие компоненты:

1. Базу данных, содержащую множество фактов;

2. Базу правил, содержащую набор продукций;

3. Механизм логического вывода или решатель;

4. ЭВМ;

5. Система общения с внешней средой.

Продукционные системы делят на 2 типа:

1. С прямым выводом, т. е. рассуждения, идут от данных к гипотезе.

2. С обратным выводом, когда вначале выдвигается гипотеза, а потом для нее ищутся доказательства.

Достоинства продукционных систем:

– модульность;

– единообразие структуры (позволяет применять оболочку продукционных систем в различных проблемных областях);

– естественность вывода знаний;

– гибкость родовидовой иерархии понятий (изменение правил влечет изменения в иерархии).

Недостатки продукционных систем:

– процесс вывода может быть менее эффективен, чем в других системах.

– процесс вывода новых знаний трудно поддается управлению.

– линейный рост объема базы знаний по мере включения новых фрагментов знаний. Если используются деревья решений, то изменения происходят по логарифмическому закону.

Другие методы представления знаний. В качестве других методов представления знаний можно указать представление знаний по примерам. При использовании такого метода база знаний заполняется следующим образом: инженер по знаниям берет задачу и заполняет матрица из совокупности задач и правильных решений. Можно построить программу с поиском аналогичной задачи (метод на основе примеров).

Такой метод применяется, например, в области медицины и в юридической практике.

Достоинством метода представления знаний по примерам является простота данного способа, т. е. для данной задачи находится метод и вносится в память.

К недостатку можно отнести отсутствие интеллектуальной гибкости.

Выводы

Центральная парадигма интеллектуальных технологий – это обработка знаний. Системы, ядром которых является база знаний или модель предметной области, описанная на языке сверхвысокого уровня, приближенном к естественному языку, называют интеллектуальными.

В связи с организацией базы знаний в информационной системе, возникли задачи по поиску оптимального и верного решения на поставленную задачу перед информационной системой. Таким образом, неструктурированную массу знаний нужно было представить в такой форме, которая позволяла бы найти оптимальное решение с наименьшими затратами, например, временными. И начиная со второй половины 20-го века в области искусственного интеллекта стали появляться различные методы представления знаний.

Рассмотренные модели представления знаний широко используются в современных интеллектуальных системах и прежде всего в экспертных системах. Каждая из форм представлений знаний может служить основой для создания языка программирования, ориентированного на работу со знаниями. В конце 80-х годов наметилась тенденция создавать комбинированные языки представления знаний. Чаще всего комбинируются фреймовые и продукционные модели.

Контрольные вопросы

1. Определение системы знаний для задач создания интеллектуальных систем.
2. Требования к системам знаний в задачах создания интеллектуальных систем.
3. Особенности знаний для их представления в компьютере.
4. Понятие инженерии знаний и баз знаний.
5. Понятие и сравнительные характеристики основных моделей представления знаний.
6. Модели представления знаний, основанные на логике предикатов, их преимущества и недостатки, области применения.
7. Семантические сети: понятие и их роль для формализации знаний.
8. Фрейм как жесткая структура информационных единиц для представления стереотипных ситуаций, достоинства и недостатки фреймовых систем.
9. Сценарии – описание стандартной последовательности фактов, определяющих типичную ситуацию предметной области.
10. Определение продукции, классификация ядер продукции, типовая схема продукционной системы (база данных, база правил, интерпретатор). Преимущества и недостатки продукционной системы.

Глава. 3. ИЗВЛЕЧЕНИЕ ЗНАНИЙ ИЗ ДАННЫХ МЕТОДАМИ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

3.1. Особенности и классификация методов интеллектуального анализа данных

Потоки текстовой и числовой информации ежедневно порождаются и оседают в хранилищах данных. Закономерности, которые скрываются в этих массивах данных, могут представлять большую ценность.

С другой стороны в связи с высокой трудоемкостью извлечения знаний по обычной технологии инженерии знаний, в последнее время интенсивно разрабатывались методы автоматического извлечения знаний из накопленных фактов. В основе этих методов лежат известные из логики методы индуктивного вывода и ряд методов распознавания образов. В западной литературе эти методы получили название «*раскопка данных и открытие знаний*» – *Data Mining (DM)* и *Knowledge Discovery in Databases (KDD)*. В отечественной литературе привился термин «Интеллектуальный анализ данных» (ИАД) [8, 19-21].

Процесс Knowledge Discovery in Databases, состоит из следующих шагов:

1. **Подготовка исходного набора данных.** Этот этап заключается в создании набора данных, в том числе из различных источников, выбора обучающей выборки и т. д. Для этого должны существовать развитые инструменты доступа к различным источникам данных. Желательно иметь поддержку работы с хранилищами данных и наличие семантического слоя, позволяющего использовать для подготовки исходных данных не технические термины, а бизнес понятия.

2. **Предобработка данных.** Для того чтобы эффективно применять методы Data Mining, следует обратить внимание на вопросы предобработки данных. Данные могут содержать пропуски, шумы, аномальные значения и т. д. Кроме того, данные могут быть избыточны, недостаточны и т. д. В некоторых задачах требуется дополнить данные некоторой априорной информацией. Наивно предполагать, что если подать данные на вход системы в существующем виде, то на выходе получим полезные знания. Данные должны быть качественны и корректны с точки зрения используемого метода DM. Поэтому первый этап KDD заключается в предобработке данных. Более того, иногда размерность исходного пространства может быть очень большой, и тогда желательно применять специальные алгоритмы понижения размерности. Это как отбор значимых признаков, так и отображение данных в пространство меньшей размерности.

3. **Трансформация, нормализация данных.** Этот шаг необходим для приведения информации к пригодному для последующего анализа виду. Для чего нужно проделать такие операции, как приведение типов, квантование, приведение к «скользящему окну» и прочее. Кроме того, некоторые методы анализа, которые требуют, чтобы исходные данные были в каком-то определенном виде. Нейронные сети, скажем, работают только с числовыми данными, причем они должны быть нормализованы.

4. **Data Mining.** На этом шаге применяются различные алгоритмы для нахождения знаний. Это нейронные сети, деревья решений, алгоритмы кластеризации, установления ассоциаций и т. д.

5. **Постобработка данных.** Интерпретация результатов и применение полученных знаний в бизнес приложениях.

Безусловно, сердцем всего этого процесса KDD являются методы DM, позволяющие обнаруживать знания.

Классической основой извлечения знаний из накопленных данных является математическая статистика. Методы математической статистики оказались полезными, главным образом, для проверки заранее сформулированных гипотез и для «грубого» разведочного анализа, составляющего основу оперативной аналитической обработки данных.

***Интеллектуальный анализ данных** — это процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей, то есть извлечения информации, которая может быть охарактеризована как знания.*

Интеллектуальный анализ данных является кратким обозначением довольно широкого спектра процедур автоматического анализа данных высокоинтеллектуальными технологиями. Эти методы позволяют извлекать из «сырых» данных ранее неизвестные зависимости между параметрами объектов и закономерности поведения классов объектов. Подобные программные продукты позволяют как бы «осмыслить» данные, оценивая их как с количественной, так и с качественной точки зрения.

Сферы применения систем Data Mining, задачи и стадии ИАД приведены в параграфе 1.4 данного пособия. Далее предлагается изложение основных методов ИАД, примеры их программной реализации для различных предметных областей в соответствии с [8, 19-21, 27].

В последнее время наряду с использованием арсенала классической статистики активно развиваются новые методы анализа данных и извлечения знаний, базирующиеся на иных, нежели традиционная интегро–дифференциальная парадигма, подходах и представленные на рис. 7:

- методы эволюционного моделирования,
- методы машинного обучения.

Термин «эволюционное моделирование» в настоящее время является достаточно устоявшимся, и общепринято под этим термином подразумевать генетические алгоритмы и искусственные нейронные сети.

Термин «машинное обучение» оставляет больше возможностей для дискуссий о том, какие методы имеются в виду, в частности, сюда относятся деревья решений.

Сформированы методологии анализа и основных принципов работы с данными в процессе анализа данных, поиска закономерностей и извлечения знаний не зависит от конкретной предметной области, поэтому может с успехом применяться для решения самого широкого спектра задач.

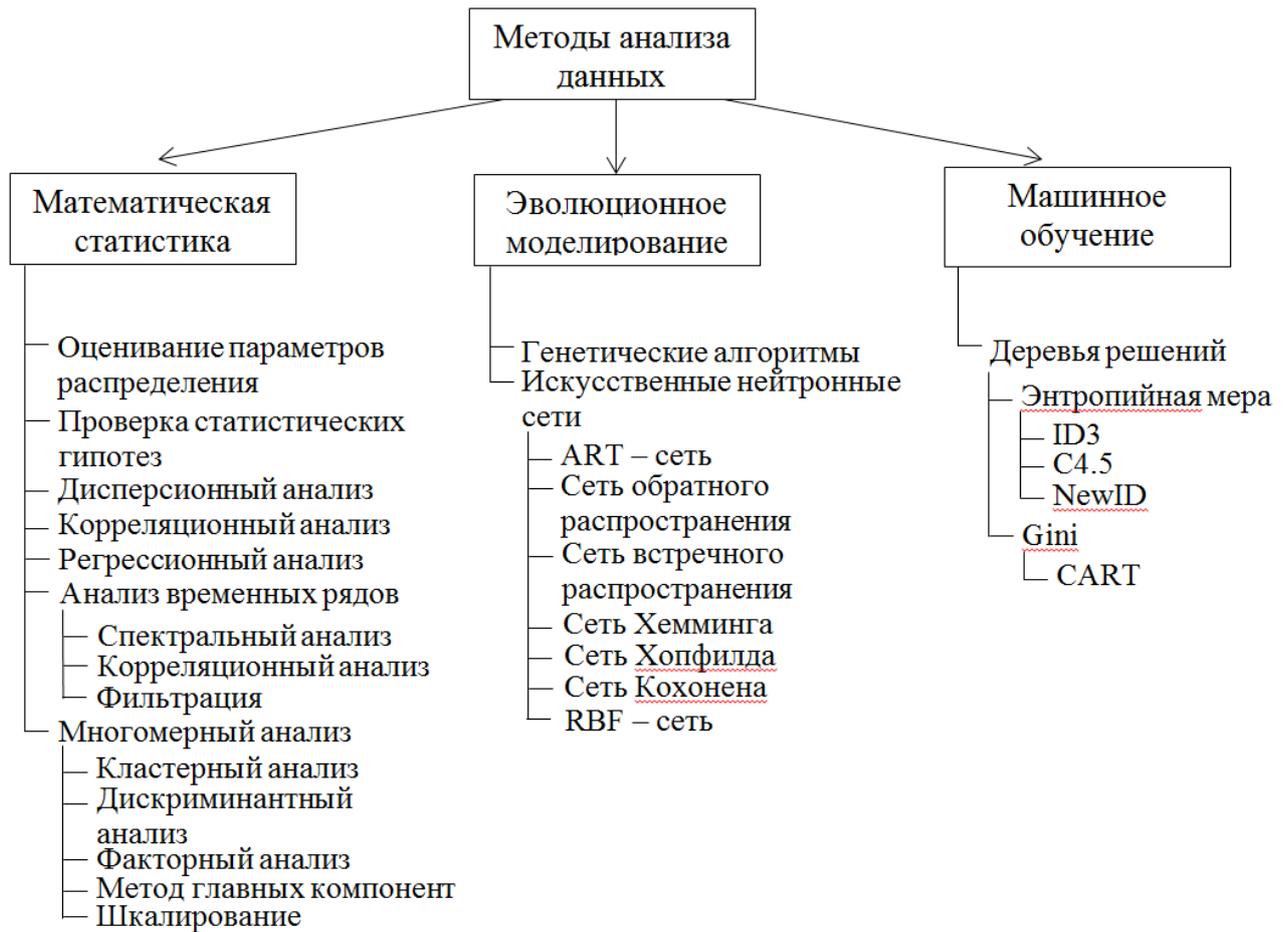


Рис.7. Классификация методов анализа данных

Алгоритмы, используемые в Data Mining, требуют большого количества вычислений. Раньше это являлось сдерживающим фактором широкого практического применения DM, однако сегодняшний рост производительности современных процессоров снял остроту этой проблемы. Теперь за приемлемое время можно провести качественный анализ сотен тысяч и миллионов записей.

Далее будут рассмотрены ряд методов интеллектуального анализа данных из числа приведенных на рис. 7. Приводится краткое изложение основных положений из статей Н.Б. Паклина и др., изложенных на сайте компании Base Group Labs – профессионального поставщика программных продуктов и решений в области анализа данных [27].

3.2. Логистическая регрессия и ROC-анализ данных

Логистическая регрессия – полезный классический инструмент для решения задачи регрессии и классификации. Без логистической регрессии и ROC-анализа, как аппарата для анализа качества моделей, немыслимо построение моделей в медицине и проведение клинических исследований. В последние годы логистическая регрессия получила распространение в скоринге для расчета рейтинга заемщиков и управления кредитными рисками. Поэтому, несмотря на

свое происхождение из статистики, логистическую регрессию и ROC-анализ почти всегда можно увидеть в наборе Data Mining алгоритмов.

Логистическая регрессия – это разновидность множественной регрессии, общее назначение которой состоит в анализе связи между несколькими независимыми переменными (называемыми также регрессорами или предикторами) и зависимой переменной. Бинарная логистическая регрессия, как следует из названия, применяется в случае, когда зависимая переменная является бинарной (т. е. может принимать только два значения). Иными словами, с помощью логистической регрессии можно оценивать вероятность того, что событие наступит для конкретного испытуемого (больной/здоровый, возврат кредита/дефолт и т. д.).

В множественной линейной регрессии предполагается, что зависимая переменная y является линейной функцией независимых переменных x_i , $i = 1, \dots, n$, т. е.:

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n.$$

Однако множественная регрессия просто игнорирует ограничения на диапазон значений для y . Поэтому для решения проблемы задача регрессии может быть сформулирована иначе: вместо предсказания бинарной переменной, мы предсказываем непрерывную переменную P со значениями на отрезке $[0,1]$ при любых значениях независимых переменных. Это достигается применением следующего регрессионного уравнения (логит-преобразование):

$$P = 1 / (1 + e^{-y}),$$

где P – вероятность того, что произойдет интересующее событие.

Поскольку логистическое преобразование решает проблему об ограничении на 0-1 границы для первоначальной зависимой переменной (вероятности), то эти преобразованные значения можно использовать в обычном линейном регрессионном уравнении. А именно, если произвести логистическое преобразование обеих частей описанного выше уравнения, мы получим стандартную модель линейной регрессии.

Существует несколько способов нахождения коэффициентов логистической регрессии. На практике часто используют метод максимального правдоподобия. Он применяется в статистике для получения оценок параметров генеральной совокупности по данным выборки. Основу метода составляет функция правдоподобия (likelihood function), выражающая плотность вероятности (вероятность) совместного появления результатов выборки y .

Кроме того известно, что логистическую регрессию можно представить в виде однослойной нейронной сети с сигмоидальной функцией активации, веса которой есть коэффициенты логистической регрессии, а вес поляризации – константа регрессионного уравнения. Как известно, однослойная нейронная сеть может успешно решить лишь задачу линейной сепарации. Поэтому возможности по моделированию нелинейных зависимостей у логистической регрессии отсутствуют. Однако для оценки качества модели логистической регрессии су-

существует эффективный инструмент ROC-анализа, что является несомненным ее преимуществом.

Для расчета коэффициентов логистической регрессии можно применять любые градиентные методы: метод сопряженных градиентов, методы переменной метрики и другие.

ROC-кривая (Receiver Operator Characteristic) – кривая, которая наиболее часто используется для представления результатов бинарной классификации в машинном обучении. Название пришло из систем обработки сигналов. Поскольку классов два, один из них называется классом с положительными исходами, второй – с отрицательными исходами. ROC-кривая показывает зависимость количества верно классифицированных положительных примеров от количества неверно классифицированных отрицательных примеров. В терминологии ROC-анализа первые называются истинно положительным, вторые – ложно отрицательным множеством. При этом предполагается, что у классификатора имеется некоторый параметр, варьируя который, мы будем получать то или иное разбиение на два класса. Этот параметр часто называют порогом, или точкой отсечения (cut-off value). В зависимости от него будут получаться различные величины *ошибок I и II рода*.

В логистической регрессии порог отсечения изменяется от 0 до 1 – это и есть расчетное значение уравнения регрессии. Будем называть его рейтингом.

Для понимания сути ошибок I и II рода рассмотрим четырехпольную таблицу сопряженности (confusion matrix), которая строится на основе результатов классификации моделью и фактической (объективной) принадлежностью примеров к классам. Таблица содержит следующие данные:

- *TP (True Positives)* – верно классифицированные положительные примеры (так называемые истинно положительные случаи);

- *TN (True Negatives)* – верно классифицированные отрицательные примеры (истинно отрицательные случаи);

- *FN (False Negatives)* – положительные примеры, классифицированные как отрицательные (ошибка I рода). Это так называемый ложный пропуск – когда интересующее нас событие ошибочно не обнаруживается (ложно отрицательные примеры);

- *FP (False Positives)* – отрицательные примеры, классифицированные как положительные (ошибка II рода). Это ложное обнаружение, т. к. при отсутствии события ошибочно выносится решение о его присутствии (ложно положительные случаи).

Что является положительным событием, а что – отрицательным, зависит от конкретной задачи.

Вводят два определения: чувствительность и специфичность модели. Ими определяется объективная ценность любого бинарного классификатора.

Чувствительность *Se (Sensitivity)* – это и есть доля истинно положительных случаев:

$$Se = TP / (TP + FN).$$

Специфичность Sp (Specificity) – доля истинно отрицательных случаев, которые были правильно идентифицированы моделью:

$$Sp = TN / (TN + FP).$$

ROC-кривая получается следующим образом:

1. Для каждого значения порога отсечения, которое меняется от 0 до 1 с шагом dx (например, 0.01) рассчитываются значения чувствительности Se и специфичности Sp . В качестве альтернативы порогом может являться каждое последующее значение примера в выборке.

2. Строится график зависимости: по оси Y откладывается чувствительность Se в процентах, по оси X откладывают $(100\% - Sp)$ (сто процентов минус специфичность), или, что тоже самое, FPR – доля ложно положительных случаев. Пример графиков приведен на рис.8.

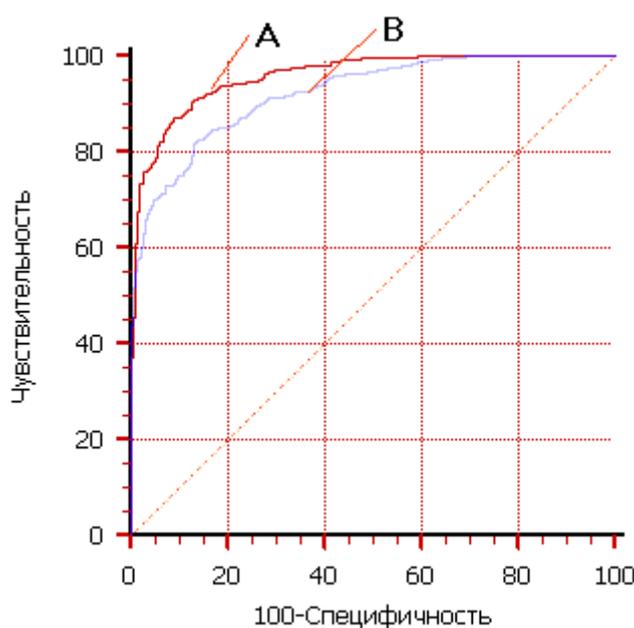


Рис. 8. – Сравнение ROC-кривых

Для идеального классификатора график ROC-кривой проходит через верхний левый угол, где доля истинно положительных случаев составляет 100% или 1.0 (идеальная чувствительность), а доля ложно положительных примеров равна нулю. Поэтому чем ближе кривая к верхнему левому углу, тем выше предсказательная способность модели. Наоборот, чем меньше изгиб кривой и чем ближе она расположена к диагональной прямой, тем менее эффективна модель. Диагональная линия соответствует бесполезному классификатору, т. е. полной неразличимости двух классов.

При визуальной оценке ROC-кривых расположение их относительно друг друга указывает на их сравнительную эффективность. Кривая, расположенная выше и левее, свидетельствует о большей предсказательной способности модели. Так, на рис. 8 две ROC-кривые совмещены на одном графике. Видно, что модель A лучше.

Идеальная модель обладает 100% чувствительностью и специфичностью. Однако на практике добиться этого невозможно, более того, невозможно одновременно повысить и чувствительность, и специфичность модели. Компромисс находится с помощью порога отсечения, так как пороговое значение влияет на соотношение Se и Sp . Можно говорить о задаче нахождения *оптимального порога отсечения* (optimal cut-off value).

Порог отсечения нужен для того, чтобы применять модель на практике: относить новые примеры к одному из двух классов. Для определения оптимального порога нужно задать критерий его определения, так как в разных задачах присутствует своя оптимальная стратегия.

3.3. Алгоритмы кластеризации на службе Data Mining

Кластеризация как объединение в группы схожих объектов является одной из фундаментальных задач в области анализа данных и Data Mining. Список прикладных областей, где она применяется, широк: сегментация изображений, маркетинг, борьба с мошенничеством, прогнозирование, анализ текстов и многие другие. Отсюда многообразие синонимов понятию кластер – класс, таксон, сгущение.

На современном этапе кластеризация часто выступает первым шагом при анализе данных.

Очень часто данные, с которыми сталкивается технология Data Mining, имеют следующие важные особенности:

- высокая размерность (тысячи полей) и большой объем (сотни тысяч и миллионы записей) таблиц баз данных и хранилищ данных (сверхбольшие базы данных);
- наборы данных содержат большое количество *числовых* и *категорийных* атрибутов.

Все атрибуты, или признаки объектов делятся на **числовые** (numerical) и **категорийные** (categorical).

Числовые атрибуты – это такие, которые могут быть упорядочены в пространстве, соответственно категориальные – которые не могут быть упорядочены. Например, атрибут «возраст» – числовой, а «цвет» – категориальный.

Большинство алгоритмов кластеризации предполагают сравнение объектов между собой на основе некоторой *меры близости* (сходства). В качестве меры близости для числовых атрибутов очень часто используется *евклидово расстояние*.

Для категориальных атрибутов распространена мера сходства Чекановского-Серенсена и Жаккара.

Можно классифицировать кластерные алгоритмы на **масштабируемые** и **немасштабируемые**.

Алгоритм называют масштабируемым (scalable), если при неизменной емкости оперативной памяти с увеличением числа записей в базе данных время его работы растет линейно.

По способу разбиения на кластеры алгоритмы бывают двух типов: *иерархические и неиерархические*.

Классические **иерархические** алгоритмы работают только с категориальными атрибутами, когда строится полное дерево вложенных кластеров. Здесь распространены агломеративные методы построения иерархий кластеров – в них производится последовательное объединение исходных объектов и соответствующее уменьшение числа кластеров. Иерархические алгоритмы обеспечивают сравнительно высокое качество кластеризации и не требуют предварительного задания количества кластеров.

Неиерархические алгоритмы основаны на оптимизации некоторой целевой функции, определяющей оптимальное в определенном смысле разбиение множества объектов на кластеры. В этой группе популярны алгоритмы семейства *k*-средних (*k*-means, fuzzy *c*-means, Густафсон-Кесселя), которые в качестве целевой функции используют сумму квадратов взвешенных отклонений координат объектов от центров искомым кластеров.

Одним из эффективных считается алгоритм обработки транзакционных данных CLOPE, который быстрее и проще в программной реализации.

Под термином *транзакция* здесь понимается некоторый произвольный набор объектов, будь это список ключевых слов статьи, товары, купленные в супермаркете, множество симптомов пациента, характерные фрагменты изображения и так далее.

Задача кластеризации транзакционных данных состоит в получении такого разбиения всего множества транзакций, чтобы похожие транзакции оказались в одном кластере, а отличающиеся друг от друга – в разных кластерах.

В основе **алгоритма кластеризации CLOPE** (англ.: Clustering with sLOPE) лежит идея максимизации глобальной функции стоимости, которая повышает близость транзакций в кластерах при помощи увеличения параметра *кластерной гистограммы*.

Пусть имеется база транзакций *D*, состоящая из множества транзакций $\{t_1, t_2, \dots, t_n\}$. Каждая транзакция есть набор объектов $\{i_1, \dots, i_m\}$. Множество кластеров $\{C_1, \dots, C_k\}$ есть разбиение множества $\{t_1, t_2, \dots, t_n\}$, такое, что $C_1 \dots C_k = \{t_1, t_2, \dots, t_n\}$ и $C_i \cap C_j = \emptyset, i, j = 1 \dots k$. Каждый элемент C_i называется *кластером*, *n*, *m*, *k* – количество транзакций, количество объектов в базе транзакций и число кластеров соответственно.

Каждый кластер *C* имеет следующие характеристики:

$D(C)$ – множество уникальных объектов; $Occ(i, C)$ – количество вхождений (частота) объекта *i* в кластер *C*; $W(C) = |D(C)|$ – ширина кластера; $H(C) = S(C)/W(C)$ – высота кластера, а площадь кластера определяется в виде $S(C) = \sum_{i \in D(C)} Occ(i, C) = \sum_{ti \in C} |ti|$.

Гистограммой кластера *C* называется графическое изображение его расчетных характеристик: по оси ОХ откладываются объекты кластера в порядке убывания величины $Occ(i, C)$, а сама величина $Occ(i, C)$ – по оси ОУ (рис. 9). На рис. 9 $S(C)$, равное 8, соответствует площади прямоугольника, ограниченного

осями координат и пунктирной линией. Очевидно, что чем больше значение H , тем более «похожи» две транзакции. Поэтому алгоритм должен выбирать такие разбиения, которые максимизируют H .

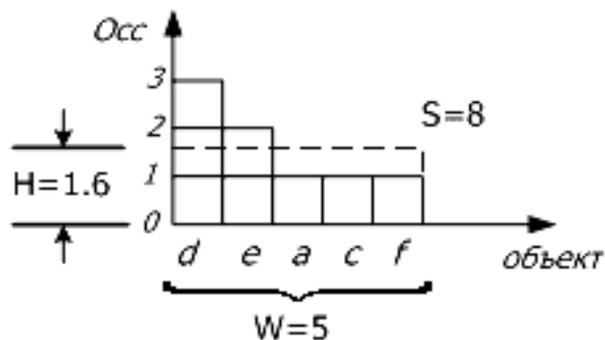


Рис.9. Пример гистограммы кластера

При совпадении значений высот применяют другие более сложные критерии: значение градиента, либо функцию стоимости.

Кластеризация транзакционных данных имеет много общего с анализом ассоциаций. Обе эти технологии Data Mining выявляют скрытые зависимости в наборах данных. Но есть и отличия.

С одной стороны, кластеризация дает общий взгляд на совокупность данных, тогда как ассоциативный анализ находит конкретные зависимости между атрибутами.

С другой стороны, ассоциативные правила сразу пригодны для использования, тогда как кластеризация чаще всего используется как первая стадия анализа.

Преимущества алгоритма CLOPE:

1. Высокие масштабируемость и скорость работы, а также качество кластеризации, что достигается использованием глобального критерия оптимизации на основе максимизации градиента высоты гистограммы кластера. Он легко рассчитывается и интерпретируется. Во время работы алгоритм хранит в RAM небольшое количество информации по каждому кластеру и требует минимальное число сканирований набора данных. Это позволяет применять его для кластеризации огромных объемов категориальных данных.

2. CLOPE автоматически подбирает количество кластеров, причем это регулируется одним единственным параметром – коэффициентом отталкивания в функции стоимости.

3.4. Деревья решений

Деревья решения являются одним из наиболее популярных подходов к решению задач Data Mining. Они создают иерархическую структуру классифицирующих правил типа «ЕСЛИ ..., ТО...», имеющую вид дерева.

Для того чтобы решить, к какому классу отнести некоторый объект или ситуацию, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня, как показано на рис.10.

Вопросы имеют вид «значение параметра A больше x ». Если ответ положительный, осуществляется переход к правому узлу следующего уровня, если отрицательный – то к левому узлу; затем снова следует вопрос, связанный с соответствующим узлом.

Введем основные понятия из теории деревьев решений, которые будут употребляться далее:

- объект – пример, шаблон, наблюдение;
- атрибут – признак, независимая переменная, свойство;
- метка класса – зависимая переменная, целевая переменная, признак определяющий класс объекта;
- узел – внутренний узел дерева, узел проверки;
- лист – конечный узел дерева, узел решения;
- проверка (test) – условие в узле.

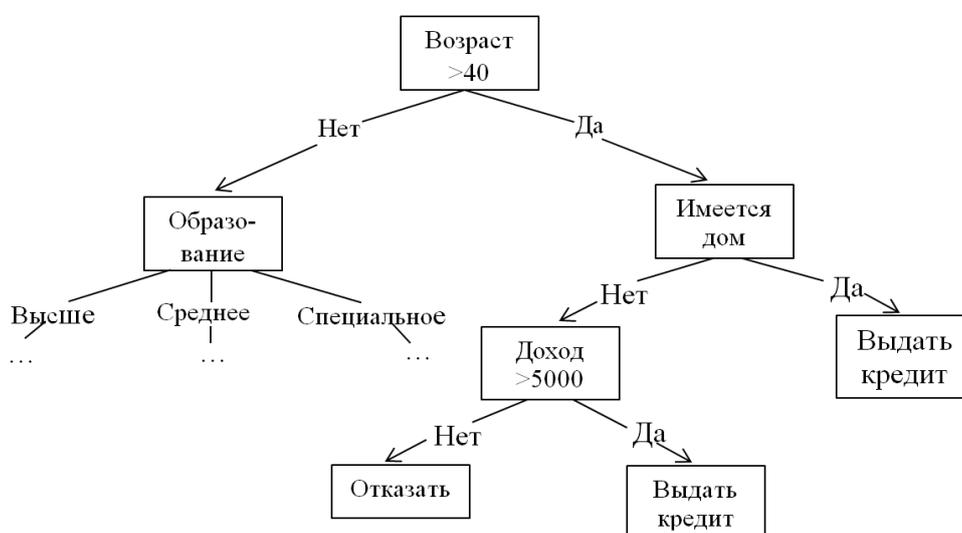


Рис.10. Пример дерева решений

Область применения дерева решений в настоящее время широка, но все задачи, решаемые этим аппаратом могут быть объединены в следующие три класса:

- **Описание данных:** Деревья решений позволяют хранить информацию о данных в компактной форме, вместо них мы можем хранить дерево решений, которое содержит точное описание объектов.
- **Классификация:** Деревья решений отлично справляются с задачами классификации, т. е. отнесения объектов к одному из заранее известных классов. Целевая переменная должна иметь дискретные значения.
- **Регрессия:** Если целевая переменная имеет непрерывные значения, деревья решений позволяют установить зависимость целевой переменной от независимых (входных) переменных. Например, к этому классу относятся задачи численного прогнозирования (предсказания значений целевой переменной).

Пусть нам задано некоторое обучающее множество T , содержащее объекты (примеры), каждый из которых характеризуется m атрибутами (атрибутами), причем один из них указывает на принадлежность объекта к определенному классу.

Пусть через $\{C_1, C_2, \dots, C_k\}$ обозначены классы (значения метки класса), тогда существуют 3 ситуации:

1. Множество T содержит один или более примеров, относящихся к одному классу C_k . Тогда дерево решений для T – это лист, определяющий класс C_k .

2. Множество T не содержит ни одного примера, т. е. пустое множество. Тогда это снова лист, и класс, ассоциированный с листом, выбирается из другого множества отличного от T , скажем, из множества, ассоциированного с родителем.

3. Множество T содержит примеры, относящиеся к разным классам. В этом случае следует разбить множество T на некоторые подмножества. Для этого выбирается один из признаков, имеющий два и более отличных друг от друга значений O_1, O_2, \dots, O_n . Множество T разбивается на подмножества T_1, T_2, \dots, T_n , где каждое подмножество T_i содержит все примеры, имеющие значение O_i для выбранного признака. Это процедура будет рекурсивно продолжаться до тех пор, пока конечное множество не будет состоять из примеров, относящихся к одному и тому же классу.

Вышеописанная процедура лежит в основе многих современных алгоритмов построения деревьев решений, этот метод известен еще под названием разделения и захвата (divide and conquer). Очевидно, что при использовании данной методики, построение дерева решений будет происходить сверху вниз.

Поскольку все объекты были заранее отнесены к известным нам классам, такой процесс построения дерева решений называется обучением с учителем (supervised learning). Процесс обучения также называют индуктивным обучением или индукцией деревьев (tree induction).

На сегодняшний день существует значительное число алгоритмов, реализующих деревья решений CART, C4.5, NewId, ITrule, CHAID, CN2 и т. д. Но наибольшее распространение и популярность получили следующие два:

– **CART (Classification and Regression Tree)** – это алгоритм построения бинарного дерева решений – дихотомической классификационной модели. Каждый узел дерева при разбиении имеет только двух потомков. Как видно из названия алгоритма, решает задачи классификации и регрессии.

– **C4.5** – алгоритм построения дерева решений, количество потомков у узла не ограничено. Не умеет работать с непрерывным целевым полем, поэтому решает только задачи классификации.

Большинство из известных алгоритмов являются жадными алгоритмами. Если один раз был выбран атрибут, и по нему было произведено разбиение на подмножества, то алгоритм не может вернуться назад и выбрать другой атрибут, который дал бы лучшее разбиение. И поэтому на этапе построения нельзя сказать даст ли выбранный атрибут, в конечном итоге, оптимальное разбиение.

При построении деревьев решений особое внимание уделяется следующим вопросам: выбору критерия атрибута, по которому пойдет разбиение, остановки обучения и отсечения ветвей. Рассмотрим все эти вопросы по порядку.

Правило разбиения. Для построения дерева на каждом внутреннем узле необходимо найти такое условие (проверку), которое бы разбивало множество, ассоциированное с этим узлом на подмножества. В качестве такой проверки должен быть выбран один из атрибутов.

Общее правило для выбора атрибута можно сформулировать следующим образом: выбранный атрибут должен разбить множество так, чтобы получаемые в итоге подмножества состояли из объектов, принадлежащих к одному классу, или были максимально приближены к этому, т. е. количество объектов из других классов (примесей) в каждом из этих множеств было как можно меньше.

Алгоритм C4.5, усовершенствованная версия алгоритма ID3 (Iterative Dichotomizer), использует теоретико-информационный подход.

Алгоритм CART использует так называемый индекс Gini (в честь итальянского экономиста Corrado Gini), который оценивает «расстояние» между распределениями классов.

Правило остановки. В дополнение к основному методу построения деревьев решений были предложены следующие правила:

- Использование статистических методов для оценки целесообразности дальнейшего разбиения, так называемая ранняя остановка (prepruning). В конечном счете ранняя остановка процесса построения привлекательна в плане экономии времени обучения.

- Ограничить глубину дерева. Остановить дальнейшее построение, если разбиение ведет к дереву с глубиной превышающей заданное значение.

- Разбиение должно быть нетривиальным, т. е. получившиеся в результате узлы должны содержать не менее заданного количества примеров.

Правило отсечения. Очень часто алгоритмы построения деревьев решений дают сложные деревья, которые переполнены данными, имеют много узлов и ветвей. Такие ветвистые деревья очень трудно понять. К тому же ветвистое дерево, имеющее много узлов, разбивает обучающее множество на все большее количество подмножеств, состоящих из все меньшего количества объектов. Ценность правила, справедливого скажем для 2-3 объектов, крайне низка, и в целях анализа данных такое правило практически непригодно. Гораздо предпочтительнее иметь дерево, состоящее из малого количества узлов, которым бы соответствовало большое количество объектов из обучающей выборки.

Для решения вышеописанной проблемы часто применяется так называемое отсечение ветвей (prepruning).

Пусть под точностью (распознавания) дерева решений понимается отношение правильно классифицированных объектов при обучении к общему количеству объектов из обучающего множества, а под ошибкой – количество неправильно классифицированных. Предположим, что нам известен способ оценки

ошибки дерева, ветвей и листьев. Тогда, возможно использовать следующее простое правило:

- построить дерево;
- отсечь или заменить поддеревом те ветви, которые не приведут к возрастанию ошибки. Отсечение ветвей происходит снизу вверх, двигаясь с листьев дерева, отмечая узлы как листья, либо заменяя их поддеревом.

Преимущества использования деревьев решений:

- быстрый процесс обучения;
- извлечение правил на естественном языке;
- генерация правил в областях, где эксперту трудно формализовать свои знания;
- интуитивно понятная классификационная модель;
- высокая точность прогноза, сопоставимая с другими методами (статистика, нейронные сети);
- построение непараметрических моделей.

В силу этих и многих других причин, методология деревьев решений является важным инструментом в работе каждого специалиста, занимающегося анализом данных, вне зависимости от того практик он или теоретик.

Деревья решений являются прекрасным инструментом в системах поддержки принятия решений, интеллектуального анализа данных. В состав многих пакетов, предназначенных для интеллектуального анализа данных, уже включены методы построения деревьев решений. В областях, где высока цена ошибки, они послужат отличным подспорьем аналитика или руководителя

Деревья решений успешно применяются для решения практических задач в следующих областях:

- Банковское дело. Оценка кредитоспособности клиентов банка при выдаче кредитов.
- Промышленность. Контроль за качеством продукции (выявление дефектов), испытания без разрушений (например, проверка качества сварки) и т.д.
- Медицина. Диагностика различных заболеваний.
- Молекулярная биология. Анализ строения аминокислот.

Это далеко не полный список областей, где можно использовать деревья решений.

3.5. Классификация программных реализаций систем Data Mining

1. Предметно-ориентированные аналитические пакеты (например, в финансовой области предназначены для предсказания курса валют). Часто основой для таких технологий является несложный статистический аппарат (аппарат прикладной статистики), в наибольшей мере учитывающий специфику рассматриваемой области. Цена таких продуктов меньше \$1000, т. е. они недорогие и их много.

2. Статистические пакеты, включающие наряду с традиционными методами математической статистики также элементы Data Mining. Круг задач, ре-

шаемых с помощью статистических пакетов: создание корреляционных моделей; факторный анализ, регрессионный анализ и др.

Достоинства данных методов состоит в том, что они основываются на строгом математическом аппарате.

Недостатки: для их использования требуется специальная математическая подготовка пользователя, а также, принципиальный недостаток пакетов состоит в том, что они опираются на статистическую парадигму и поэтому дают математически усредненный вывод.

Наиболее распространены такие мощные статистические пакеты, как SAS (компания SAS Institute), SPSS (SPSS), Statistica и др. Эти системы достаточно дороги – от \$1000 до \$15000.

3. Нейросетевые системы. Принцип работы нейронных сетей заключается в следующем: имитируется работа нейронов в составе иерархической сети. При этом сигналы передаются от нижних слоев к верхним, а сила сигнала зависит от сложности связи.

Недостатки:

- необходимо иметь большой объем обучающей выборки;
- результаты работы такого пакета трудно объяснить.

Нейронные сети могут быть реализованы программным или аппаратным способом.

Вариантами аппаратной реализации являются нейрокомпьютеры, нейроплаты и нейроБИС (большие интегральные схемы). Одна из самых простых и дешёвых нейроБИС – модель MD 1220 фирмы Micro Devices, которая реализует сеть с 8 нейронами и 120 синапсами.

Большинство современных нейрокомпьютеров представляют собой персональный компьютер или рабочую станцию, в состав которых входит дополнительная нейроплата. К их числу относятся, например, компьютеры серии FMR фирмы Fujitsu.

К программам, реализующим технологии нейронных сетей относятся: пакет Brain Maker (CSS), NeuroShell (Ward Systems Group) и др. Стоимость таких систем очень высока (более \$1000). Пакет Brain Maker используется несколькими тысячами финансовых и промышленных компаний, а также оборонными ведомствами США для решения задач прогнозирования, оптимизации и моделирования ситуаций. К таким задачам относятся прогнозирование курсов валют и акций на биржах, моделирование кризисных ситуаций, распознавание образов и многие другие на основе использования сети Хопфилда с обучением по методу обратного распространения ошибки.

4. Системы рассуждений на основе аналогичных случаев (Case Based Reasoning – CBR) основаны на поиске аналога наличной ситуации. Назначение таких систем: среди БД множества случаев найти тот, который наиболее близок к рассматриваемому.

Недостатки:

- не создается никакой модели (ищем аналогию, но не ищем модель, правило);

– результат зависит от меры близости, которую можно выбрать и достаточно неправильной. Для эффективного решения задач классификации или прогноза необходимо, чтобы был накоплен большой объем данных.

Пример системы, использующей CBR, – Pattern Recognition Workbench (Unisca, США).

5. Деревья решений (Decision Tree). Указанная технология используется в том случае, если применимо правило типа «If ...,Then ...» и она представляет собой процесс мышления, при котором идет логический вывод от корня к ветвям и листьям. Такие переходы понятны и видны графически, т. е. пользователь получает модель, по которой он может сделать выводы о результатах.

Недостаток технологии деревьев решений заключается в том, что трудно найти оптимальное решение, такая система не дает ответа, какое решение следует считать оптимальным.

На использовании технологии деревьев решений работают продукты See5 / C5.0 (компания RuleQuest), KnowledgeSeeker (ANGOSS) и др. Стоимость таких пакетов от \$1000.

6. Пакеты на основе алгоритмов ограниченного перебора. Суть работы таких технологий заключается в следующем: алгоритм вычисляет частоты комбинаций простых логических событий в подгруппах данных и на основании вычисленных частот делается вывод о полезности той или иной комбинации. Они определяют логические «If ...,Then ...» правила и выдают результат. Технология ограниченного перебора используется там, где трудно найти функциональную связь.

Известным представителем этого подхода является продукт WizWhy (фирма WizSoft). Стоимость таких пакетов – около \$4000.

7. Генетические алгоритмы. Данная технология представляет собой попытку найти решение путем обращения к генетике. Генетический алгоритм (ГА) – это поисковый алгоритм, основанный на природных механизмах селекции и генетики. Эти алгоритмы обеспечивают выживание сильнейших решений из множества сгенерированных, формируя и изменяя процесс поиска на основе моделирования эволюции исходной популяции решений.

Коммерческое программное обеспечение, реализующее ГА, можно разделить на программные средства общего назначения, прикладные и алгоритмические программные продукты.

Программное обеспечение общего назначения включает разнообразные наборы инструментальных средств для построения конкретных программ, которые содержат библиотеки алгоритмов, программы моделирования, средства визуализации и другие инструменты. Пакеты подобного типа рассчитаны на опытных программистов, требуют знания основ теории эволюционных вычислений и характеризуются высокой трудоёмкостью освоения, которая в значительной мере зависит от квалификации пользователя.

Прикладные программные продукты ориентированы на решение проблем определённого класса в конкретных предметных областях (реинжиниринг, маркетинг, стратегическое планирование и др.). Такие средства не требуют от

пользователя теоретических знаний в области методологии создания интеллектуальных систем. Достаточно, чтобы он был специалистом в своей предметной области.

Алгоритмическое программное обеспечение поддерживает один (или несколько) генетический алгоритм. Преимущества таких программных продуктов – их гибкость и простота использования. При этом пользователям необходимо иметь представление об основах теории ГА.

Система PC/Beagle представляет собой программу поиска решающих правил, классифицирующих примеры из базы данных. Она превращает данные в знания за счёт использования машинного обучения. Один из модулей системы путём репродукции и селекции порождает правила, представленные в виде логических выражений.

Система Evolver реализует шесть методов генетической оптимизации и выполнена в виде расширения MS Excel. Основные области применения пакета – оптимизация доходности с учётом уровня риска и максимизация прибыли с учётом возможных издержек.

Genesis – известный алгоритмический программный продукт, который используется в качестве инструмента тестирования генетических алгоритмов. Он позволяет создать модифицированную программную среду и обеспечивает пользователя статистической информацией на выходе.

Программный продукт общего назначения EnGENEer помогает адаптировать генетические алгоритмы к новым проблемным областям.

Известен пакет GeneHunter фирмы Ward Systems Group, который может использоваться как приложение MS Excel и допускает составление собственных программ на языках C и Visual Basic. Его стоимость – около \$1000.

8. Эволюционное программирование. Процесс построения программ строится как эволюции в мире программ. Когда система находит программу, более или менее выражающую искомую зависимость, она начинает вносить в нее небольшие модификации и отбирает среди построенных дочерних программ те, которые повышают точность. Так работает отечественная программная система PolyAnalyst.

Другое направление эволюционного программирования связано с поиском зависимости целевых переменных от остальных в форме функций какого-то определенного вида, например, в форме полиномов. Указанное направление реализовано в системе NeuroShell компании Ward System Group. Стоимость систем до \$4000.

9. Аналитические платформы. Аналитические программные комплексы имеют тенденцию совершенствоваться в направлении увеличения количества параметров многомерного анализа и уровня адекватности математического аппарата реальным, многосторонним бизнес-процессам для повышения точности и долгосрочности прогнозирования развития процессов и ситуаций, и их оптимизации в интересах предприятий и организаций. Аналитические программные комплексы должны создаваться с расчетом на слияние с другими интеллектуальными средствами, имитационными математическими моделями, многокри-

териальными системами поддержки принятия решений, системами искусственного интеллекта и др. Только в этом случае возможно образование интегрированных комплексов поддержки принятия оптимальных решений, причем как в интересах достижения стратегических целей, так и в интересах своевременного, достаточно быстрого и адекватного реагирования на изменения внешних условий в процессе управления деятельностью предприятий и организаций.

Разработанный в России компанией BaseGroup Labs программный продукт Deductor является аналитической платформой, т. е. основой для создания законченных прикладных решений на основе решения задач анализа данных. Реализованные в Deductor технологии предоставляет полный набор инструментов и позволяют на базе единой архитектуры пройти все этапы построения аналитической системы: от создания хранилища данных до автоматического подбора моделей и визуализации полученных результатов [21, 27].

Deductor предоставляет аналитикам инструментальные средства, необходимые для решения самых разнообразных аналитических задач: корпоративная отчетность, прогнозирование, сегментация, поиск закономерностей – эти и другие задачи, где применяются такие методики анализа, как OLAP, Knowledge Discovery in Databases и Data Mining. Deductor является идеальной платформой для создания систем поддержки принятия решений.

3.6. Краткое описание применения аналитической платформы Deductor для решения задач интеллектуального анализа данных

Состав платформы Deductor. Система состоит из пяти частей:

– **Studio** – программа, реализующая функции импорта, обработки, визуализации и экспорта данных. Deductor Studio может функционировать и без хранилища данных, получая информацию из любых других источников, но наиболее оптимальным является их совместное использование. В Deductor Studio включен полный набор механизмов, позволяющий получить информацию из произвольного источника данных, провести весь цикл обработки (очистку, трансформацию данных, построение моделей), отобразить полученные результаты наиболее удобным образом (OLAP, диаграммы, деревья и др.) и экспортировать результаты на сторону.

– **Viewer** – рабочее место конечного пользователя. Позволяет отделить процесс построения сценариев от использования уже готовых моделей. Все сложные операции по подготовке сценариев обработки выполняются аналитиками-экспертами при помощи Deductor Studio, а Deductor Viewer обеспечивает пользователям простой способ работы с готовыми результатами, скрывает от них все сложности построения моделей и не предъявляет высоких требований к квалификации сотрудников.

– **Warehouse** – многомерное хранилище данных, аккумулирующее всю необходимую для анализа предметной области информацию. Использование единого хранилища позволяет обеспечить непротиворечивость данных, их централизованное хранение и автоматически обеспечивает всю необходимую поддержку процесса анализа данных.

– **Server** – служба, обеспечивающая удаленную аналитическую обработку данных. Позволяет автоматически обрабатывать данные и переобучать модели на сервере, оптимизирует выполнение сценариев за счет кэширования проектов и использования многопоточной обработки.

– **Client** – клиент доступа к Deductor Server. Обеспечивает доступ к серверу из сторонних приложений и управление его работой.

Реализованная в Deductor архитектура позволяет добиться максимальной гибкости при создании законченного решения. Благодаря данной архитектуре можно собрать в одном аналитическом приложении все необходимые инструменты анализа и реализовать автоматическое выполнение подготовленного сценария. Создание законченного решения занимает очень мало времени. Достаточно получить данные, определить сценарий обработки и задать место для экспорта полученных результатов. Наличие мощного набора механизмов обработки и визуализации позволяет двигаться по шагам, от наиболее простых способов анализа к более мощным. Таким образом, первые результаты пользователь получает практически сразу, но при этом можно легко наращивать мощность решения.

Хранилище данных. Deductor включает в себя кросс-платформенное хранилище Deductor Warehouse – специализированную базу, в которую загружается информация из существующих учетных систем. Оно спроектировано специально для обработки аналитических запросов и включает удобный семантический слой, благодаря которому пользователи в состоянии извлечь любые интересующие их сведения без написания сложных запросов, только оперируя понятными бизнес-терминами.

Deductor поддерживает работу практически с любыми источниками данных: офисные приложения, 1С:Предприятие, популярные СУБД, ERP-, CRM-, BI-системы, текстовые файлы, XML-источники и десятки других. Где бы ни хранилась информация, ее можно будет извлечь, трансформировать к нужному виду и загрузить в хранилище данных.

Интеграция. Для получения аналитической отчетности при помощи Deductor может использоваться информация из десятков источников: хранилища данных, учетные системы, СУБД, файлы и многое другое. Полученные результаты можно просмотреть на экране, передать в офисные приложения (MS Excel, Word, Access), разместить на веб-сайте, загрузить в базу данных.

Простота применения. Для построения аналитического решения на базе платформы Deductor не требуется написания сложного кода. Практически все операции производятся в режиме визуального проектирования, при помощи специальных мастеров, облегчающих работу аналитика, которая отделена от работы конечного пользователя. Аналитик строит при помощи мастеров сценарии, в которых реализована произвольная логика обработки, а конечный пользователь просто выбирает нужный отчет и автоматически получает интересующие данные.

Прогнозирование – одна из самых востребованных, но при этом и самых сложных задач анализа. Качественный прогноз является ключом к решению та-

ких актуальных бизнес-задач, как оптимизация складских запасов и финансовых потоков, бюджетирование, оценка инвестиционной привлекательности и многие другие. Обработчики в Deductor обеспечивают возможность построения прогностических моделей с использованием как простых моделей (по среднему, на основе предыдущих периодов, по жестким правилам), так и мощных самообучающихся алгоритмов (регрессионные модели, нейронные сети, анализ временных рядов).

OLAP-визуализатор. Результаты планирования, моделирования, различные варианты сценарных прогнозов можно просмотреть при помощи множества удобных визуализаторов, в частности с использованием встроенного OLAP-модуля. Он поддерживает весь необходимый для бюджетирования функционал: просмотр в любых срезах, детализация (drill down), сортировки, группировки, множество способов фильтрации.

Анализ отклонений. Одним из важнейших этапов работы системы бюджетирования является выявление отклонений и анализ причин их возникновения. Deductor включает мощные Data Mining алгоритмы, позволяющие не только найти аномалии, отклонения, противоречия, но и проанализировать что привело к такому результату, выявить влияние на аномалии множества факторов, найти причинно-следственные связи.

Версии платформы. Существует три версии этой платформы. Версия *Enterprise* предназначена для корпоративного использования. В ней присутствуют:

- серверные компоненты Deductor Server и Deductor Client;
- интерфейс доступа к Deductor через механизм OLE Automation;
- традиционное хранилище данных Deductor Warehouse на трех СУБД: Firebird, MS SQL, Oracle;
- виртуальное хранилище данных Deductor Virtual Warehouse.

Версия *Professional* предназначена для небольших компаний и однопользовательской работы. В ней отсутствуют серверные компоненты, поддержка OLE, виртуальное хранилище, а традиционное хранилище данных можно создавать только на СУБД Firebird.

Версия *Academic* предназначена для образовательных и обучающих целей. Ее функционал аналогичен версии Professional за исключением того, что отсутствуют пакетный запуск сценариев, т. е. работа в программе может вестись только в интерактивном режиме, а также нет импорта из промышленных источников данных: 1С, СУБД, файлы MS Excel, Deductor Data File.

Реализованные технологии Data Mining:

- задача ассоциации на основе алгоритма Apriori;
- задача кластеризации на основе сетей и карт Кохонена, а также алгоритмов k-means и G-means;
- задача классификации и регрессии на основе линейной и логистической регрессии, деревьев решений (алгоритм C 4.5), многослойного персептрона (алгоритмы BProp, RProp);

- анализ и прогнозирование временных рядов на основе регрессионных и нейросетевых моделей;
- сравнение моделей на основе ROC-анализа.

В Deductor Studio вся работа ведется с использованием пяти мастеров.

Для создания сценариев обработки данных используются *мастера импорта, обработки и экспорта*. Для настройки подключений к источникам данных используется *мастер подключений*. *Мастер визуализации* настраивает визуализаторы для конкретного узла.

Подробное изложение системы Deductor изложено в работах Н.Б. Паклина и других разработчиков и исследователей в [21, 27]. Учебные примеры применения платформы содержатся также в пособиях [22, 23].

Сопоставление Deductor со известными специализированными продуктами других производителей. По производительности работа Warehouse на Oracle, SQL и Firebird при более-менее активном лазании в Хранилище (закачка транзакций и какой-то дополнительной информации, а также постоянное ее использование для прогнозирования порядка 30000-40000 временных рядов в месяц и сохранение прогнозов) отличается существенно, при больших объемах раз в 10-20. При грамотной оптимизации можно добиться еще большей разницы в скорости. Это связано с тем, что тяжелые базы сами по себе работают быстрее (точнее могут больше выжать из оборудования), но не только. Дело в том, что у Oracle и MS SQL есть нестандартные опции, позволяющие загружать данные пачками, извлекать большими кусками и прочее. Кроме того, поддерживается размещение в различные табличные пространства и прочее. При небольших объемах (база до гигабайта) разницы в производительности у указанных баз данных и Deductor 5 почти нет [27]. В Deductor 5 есть алгоритмы для построения прогнозных моделей. Так что теоретических проблем нет, а вот на счет качества – это уже вопрос обучения, данных и прочее. Для работы нужен Deductor Studio. Даже бесплатная версия Deductor Academic может обрабатывать выборки объемом в несколько миллионов записей без особых проблем.

Выводы

Технологии извлечения знаний из хранилищ данных основаны на методах статистического анализа и моделирования, ориентированных на поиск моделей и отношений, скрытых в совокупности данных. Эти модели могут в дальнейшем использоваться для оптимизации деятельности предприятия или фирмы.

Для извлечения значимой информации из хранилищ данных имеются специальные методы (OLAP-анализа, Data Mining или Knowledge Discovery), основанные на применении методов математической статистики, нейронных сетей, индуктивных методов построения деревьев решений и др.

Контрольные вопросы

1. Каковы основные шаги процедур Knowledge Discovery in Databases при работе с данными в процессе анализа данных?
2. Перечислите основные методы из арсенала математической статистики, применяемые при анализе данных.

3. Перечислите и кратко охарактеризуйте основные методы из арсенала эволюционного моделирования, применяемые для интеллектуального анализе данных.

4. Перечислите и кратко охарактеризуйте основные методы из арсенала машинного обучения, применяемые для интеллектуального анализе данных.

5. Дайте определение бинарной логистической регрессии, логит-преобразования, определение ROC-кривой для представления результатов бинарной классификации в машинном обучении.

6. Какова процедура построения и визуальной оценки ROC-кривых для представления результатов бинарной классификации и прогноза?

7. Каковы особенности алгоритмов кластеризации наборов данных, содержащих большое количество числовых и категориальных атрибутов?

8. В чем смысл задачи кластеризации транзакционных данных?

9. Какие процедуры содержит алгоритм кластеризации транзакционных данных CLOPE?

10. Каковы преимущества алгоритма кластеризации транзакционных данных CLOPE?

11. Каковы основные понятия из теории деревьев решений используются при решении задач Data Mining?

12. Каковы области применения дерева решений при решении задач Data Mining?

13. Какие процедуры лежат в основе многих современных алгоритмов построения деревьев решений?

14. Каковы особенности алгоритмов CART и C4.5, реализующих деревья решений?

15. Выбор критерия атрибута, по которому пойдет разбиение, остановка обучения и отсечение ветвей деревьев решений.

16. Перечислите и кратко охарактеризуйте виды программных реализаций систем Data Mining.

17. Каковы параметры алгоритма конструирования классификатора в программе See5?

18. Как преобразовать полученное дерево решений в набор правил If...Then?

19. Как можно уменьшить ошибки классификации в программе See5?

20. Какой алгоритм положен в основу работы программы WizWhy для поиска логических правил в данных?

21. Какие параметры следует задать для работы процедуры поиска правил и какие отчеты готовит система WizWhy?

22. Каков перечень решаемых задач и состав аналитической платформы Deductor 5?

23. В чем заключена новизна взгляда на данные при применении аналитической платформы Deductor?

24. Какой объем выборки данных может обрабатывать и какие задачи может решать демоверсия Deductor 5?

ЗАКЛЮЧЕНИЕ

В сферу приложений искусственного интеллекта в настоящее время вошли практически все направления современной информатики. Многие из этих направлений, таких как вопросы создания экспертных систем, искусственных нейросетей и др., в достаточной мере освещены в учебной и методической литературе. Поэтому я не ставил перед собой цель «объять необъятное», а включил в текст лекций как разделы, содержащие описание традиционных моделей знаний, понятий, определений и типов интеллектуальных систем, так и новый перспективный подход к решению проблем интеллектуальной обработки накопленных массивов данных.

При написании пособия автор видел свою цель в том, чтобы познакомить студентов с принципами создания и функционирования интеллектуальных информационных систем. Практический опыт показывает, что люди, не знакомые с этими принципами, испытывают большие трудности, выступая в роли пользователей интеллектуального программного обеспечения.

С учетом динамичного развития современной области ИИ текст лекций является лишь введением в огромную и интересную область знаний. Слушатели, заинтересовавшиеся работами в этом перспективном научном направлении и вопросами практического применения методов ИИ в решении сложных задач управления, имеют прекрасную возможность самостоятельно продолжить изучение этой тематики. Огромный мир Интернета позволяет получить быстрый доступ к самой различной информации по тематике создания и применения интеллектуальных систем и технологий, к имеющейся обширной библиографии, углубленным теоретическим и практическим курсам по систематическому освещению многих вопросов, относящихся к области ИИ и ИС. Многие перспективные направления ИИ, ИС в работе лишь упомянуты, или приведены в качестве примеров развития возможных теоретических направлений и практических применений.

Автор надеется, что знания, полученные при изучении данной книги, помогут читателю решать практические примеры при самостоятельном освоении технологии и продуктов Data Mining, а также позволят расширить кругозор в области интеллектуальных информационных систем и технологий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Искусственный интеллект: справочник. В 3 кн. Кн. 1: Системы общения и экспертные системы / под ред. Э.В. Попова. – М.: Радио и связь, 1990. – 461 с.
2. Искусственный интеллект: справочник. В 3 кн. Кн. 2: Модели и методы / под ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – 304 с.
3. Искусственный интеллект: справочник. В 3 кн. Кн. 3: Программные и аппаратные средства / под ред. В.Н. Захарова и В.Ф. Хорошевского. – М.: Радио и связь, 1990. – 320 с.
4. Коршунов, Ю.М. Математические основы кибернетики: учебное пособие для вузов / Ю.М. Коршунов. – М.: Энергоатомиздат, 1987. – 496 с.
5. Острейковский, В.А. Информатика: учебник для вузов / В.А. Острейковский. – М.: Высшая школа, 2009. – 510 с.
6. Тельнов, Ю.Ф. Интеллектуальные информационные системы: учебное пособие / Ю.Ф. Тельнов. – М.: Московский международный институт эконометрики, информатики, финансов и права, 2002. – 118 с.
7. Интеллектуальные информационные системы и технологии: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. – Тамбов: Изд-во ТГТУ, 2013. – 244 с.
8. Дюк, В. Data Mining: учебный курс / В. Дюк, А. Самойленко. – СПб.: Питер, 2001. – 366 с.
9. Попов, Э.В. Статические и динамические экспертные системы / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель и др. – М.: Финансы и статистика, 1997. – 310 с.
10. Логиновский, О.В. Управление и стратегии: учебное пособие / О.В. Логиновский. – Оренбург: Изд-во ОГУ; Челябинск: Изд-во ЮУрГУ, 2001. – 704 с.
11. Луценко, Е.В. Интеллектуальные информационные системы: учебное пособие для студентов специальности 230400 «Информационные системы и технологии» / Е.В. Луценко. – Краснодар: КубГАУ, 2013. – 645 с.
12. Гаврилова, Т.А. Базы знаний интеллектуальных систем: учебное пособие для вузов / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000. – 382 с.
13. Романов, В.П. Интеллектуальные информационные системы в экономике: учебное пособие для вузов по специальности «Прикладная информатика» и другим междисциплинарным специальностям / под общ. ред. Н.П. Тихомирова. – М.: Экзамен, 2003. – 494 с.
14. Макаренко, С.И. Интеллектуальные информационные системы: учебное пособие / С.И. Макаренко. – Ставрополь: СФ МГГУ, 2009. – 206 с.
15. Таунсенд, К. Проектирование и программная реализация экспертных систем на персональных ЭВМ / К. Таунсенд, Д. Фохт; пер. с англ. – М.: Финансы и статистика, 1990. – 356 с.
16. Программные средства интеллектуальных систем / А.Е. Городецкий. – СПб.: Изд. СПбГТУ, 2000. – 316 с.

17. Бессмертный, И.А. Искусственный интеллект: учебное пособие / И.А. Бессмертный. – СПб: СПбГУ ИТМО, 2010. – 132 с.
18. Барыкин, С.Г. Системы искусственного интеллекта: учебное пособие / С.Г. Барыкин, Н.В. Плотникова. – Челябинск: Изд-во ЮУрГУ, 2004 – 85 с.
19. Барсегян, А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP: учебное пособие для специальности «Информационные системы» / А.А. Барсегян. – СПб.: БХВ Петербург, 2007. – 376 с.
20. Кацко, И.А. Практикум по анализу данных на компьютере: учебное пособие / И.А. Кацко, Н.Б. Паклин. – М.: КолосС, 2009.– 278 с.
21. Паклин, Н.Б. Бизнес-аналитика: от данных к знаниям: учебное пособие / Н.Б. Паклин, В.И. Орешков. – СПб.: Питер, 2013. – 704 с.
22. Поллак, Г.А. Современные технологии анализа информации: учебное пособие / Г.А. Поллак. – Челябинск: Издательский центр ЮУрГУ, 2013 – 115 с.
23. Поллак, Г.А. Современные технологии анализа информации: учебное пособие к практическим работам / Г.А. Поллак. – Челябинск: Издательский центр ЮУрГУ, 2013 – 99 с.
24. Чинакал, В.О. Интеллектуальные системы и технологии: учебное пособие / В.О. Чинакал. – М.: РУДН, 2008. – 303 с.
25. Коровин, А.М. Управление знаниями на основе ИТ-технологий: текст лекций / А.М. Коровин. – Челябинск: Издательский центр ЮУрГУ, 2013. – 48 с.
26. <http://www.ipu.ru/labs/lab51/projects.html>.
27. <http://www.basegroup.ru/deductor/>
28. <http://www.rulequest.com/>
29. <http://www.interface.ru/sysmod/>
30. <http://www.bipartner.ru/services/dm.html>.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Глава 1. ПОНЯТИЯ И ТЕРМИНОЛОГИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	
1.1. История исследований и основные понятия в области искусственного интеллекта.....	5
1.2. Основные направления исследований в области интеллектуальных систем	8
1.3. Основные признаки и отличия интеллектуальных систем.....	10
1.4. Основные типы интеллектуальных систем.....	13
Глава 2. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ	
2.1. Представление знаний, рассуждений и задач.....	28
2.2. Модели представления знаний.....	30
Глава 3. ИЗВЛЕЧЕНИЕ ЗНАНИЙ ИЗ ДАННЫХ МЕТОДАМИ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ	
3.1. Особенности и классификация методов интеллектуального анализа данных	36
3.2. Логистическая регрессия и ROC-анализ данных.....	38
3.3. Алгоритмы кластеризации на службе Data Mining	42
3.4. Деревья решений.....	44
3.5. Классификация программных реализаций систем Data Mining.....	48
3.6. Краткое описание применения аналитической платформы Deductor для решения задач интеллектуального анализа данных.....	52
ЗАКЛЮЧЕНИЕ.....	57
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	58

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



И. о. проректора по учебно-методической работе
В. В. Зубов

УТВЕРЖДАЮ

**Методические указания
к выполнению лабораторных работ
Часть 1**

**Б1.В.09.08 МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В МЕХАТРОНИКЕ И
РОБОТОТЕХНИКЕ**

Специальность
21.05.04 Горное дело

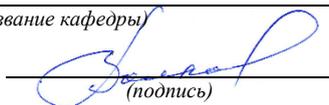
Специализация
Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой


(подпись)

Волков Е. Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

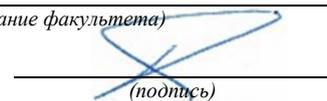
(Дата)

Рассмотрены методической комиссией
факультета

Горно-механического

(название факультета)

Председатель


(подпись)

Осипов П. А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

ОГЛАВЛЕНИЕ

1. ОБЩЕЕ ОПИСАНИЕ СТЕНДА	4
1.1 Назначение стенда	4
1.2 Состав.....	5
1.3 Технические характеристики стенда	5
2. ЛАБОРАТОРНЫЕ РАБОТЫ.....	6
2.1 Работа №1. Изучение контроллера ОВЕН ПЛК-160.....	6
2.2 Работа №2. Основы работы со SCADA-системой.....	42

1. ОБЩЕЕ ОПИСАНИЕ СТЕНДА

1.1 Назначение стенда

Лабораторный стенд предназначен для изучения программируемого логического контроллера с объектом управления в виде мехатронного модуля – сборки деталей. Стенд позволяет изучить основы построения систем цикловой автоматики. Элементная база мехатронного модуля состоит из ленточного конвейера с электроприводом постоянного тока, модуля подачи базовых деталей, модулей подачи крышек из различного материала, пневматических толкателей и датчиков различного типа.

Внешний вид учебного комплекта представлен ниже (см. Рисунок 1).



Рисунок 1 – Внешний вид стенда ММ-СБ-НН

1.2 Состав

Лабораторный стенд включает в себя следующие компоненты (см. Таблица 1).

Таблица 1

№	Наименование	Кол-во
1	Моноблок, содержащий: источник питания, программируемый логический контроллер, элементы индикации и управления	1 шт.
2	Мехатронный модуль сборки деталей	1 шт.
3	Модуль пульта симуляции	1 шт.
4	Компрессор	1 шт.
5	Ноутбук	1 шт.
7	Комплект кабелей и принадлежностей для проведения лабораторных работ	1 шт.
8	Техническое описание	1 шт.
9	Методические указания к проведению лабораторных работ	1 шт.

1.3 Технические характеристики стенда

Общие технические характеристики типового комплекта учебного оборудования представлены ниже (см. Таблица 2).

Таблица 2

Параметр	Значение
Напряжение электропитания, В	220
Частота питающего напряжения, Гц	50
Потребляемая мощность, ВА	1600
Габаритные размеры, мм	480x600x650
Масса, кг	60 кг
Диапазон рабочих температур	+10...+35 °С
Относительная влажность воздуха, не более	80%

2. ЛАБОРАТОРНЫЕ РАБОТЫ

2.1 Работа №1. Изучение контроллера ОВЕН ПЛК-160

Цель работы

Научиться составлять простейшие программы на программируемом логическом контроллере ОВЕН

Порядок работы

1. Изучить необходимый теоретический материал.
2. Ознакомиться с конструкцией и назначением элементов лабораторного стенда «Промышленные датчики давления».
3. Подготовить стенд к проведению лабораторной работы.
4. Изучить назначение, технические характеристики ОВЕН ПЛК-160.
5. Изучить основы системы команд и принципы программирования контроллера.
6. Дома при подготовке к работе:
 - выполнить синтез системы автоматизации согласно варианту;
 - составить программу на языке лестничных диаграмм.
7. В лаборатории:
 - освоить графическую среду программного обеспечения CODESYS;
 - набрать на компьютере подготовленную дома программу и откомпилировать ее;
 - ввести программу в контроллер и убедиться в правильности её работы.

Ход работы

В данной лабораторной работе программирование пользовательской задачи реализуется при помощи программного обеспечения CODESYS.

Обучающийся, в соответствии с поставленной задачей, разрабатывает алгоритм автоматизации управления объектом либо в виде логических уравнений, либо в виде схемы алгоритма. Входным, выходным сигналам и внутренним переменным (меркерам) присваиваются адреса в соответствии с конфигурационной таблицей.

Перед проведением лабораторной работы необходимо установить все элементы стенда в исходное состояние. Для этого при выключенном автоматическом выключателе QF1 «Сеть», расположенном на лицевой панели стенда:

- установить в выключенное состояние клавишный переключатель «Питание контроллера»;
- тумблеры блока дискретных входов I1 ... I8 перевести в нижнее положение, соответствующее состоянию «Выкл»;

– рукоятку потенциометра PR1 блока аналогового ввода/вывода перевести в крайнее положение против часовой стрелки;

После установки начальных условий необходимо подготовить к работе персональный компьютер и обеспечить его связь со стендом:

– включить персональный компьютер и дождаться загрузки операционной системы Windows;

– подать напряжение на стенд включением автоматического выключателя QF1 «Сеть»;

– подать напряжение на программируемый логический контроллер и датчики включением клавишного выключателя «Питание контроллера», дождаться загрузки ПЛК.

В комплексе используется программируемый логический контроллер ОВЕН ПЛК-160, внешний вид которого представлен на рис. 3. Описание основных элементов ПЛК представлено в таблице 1.

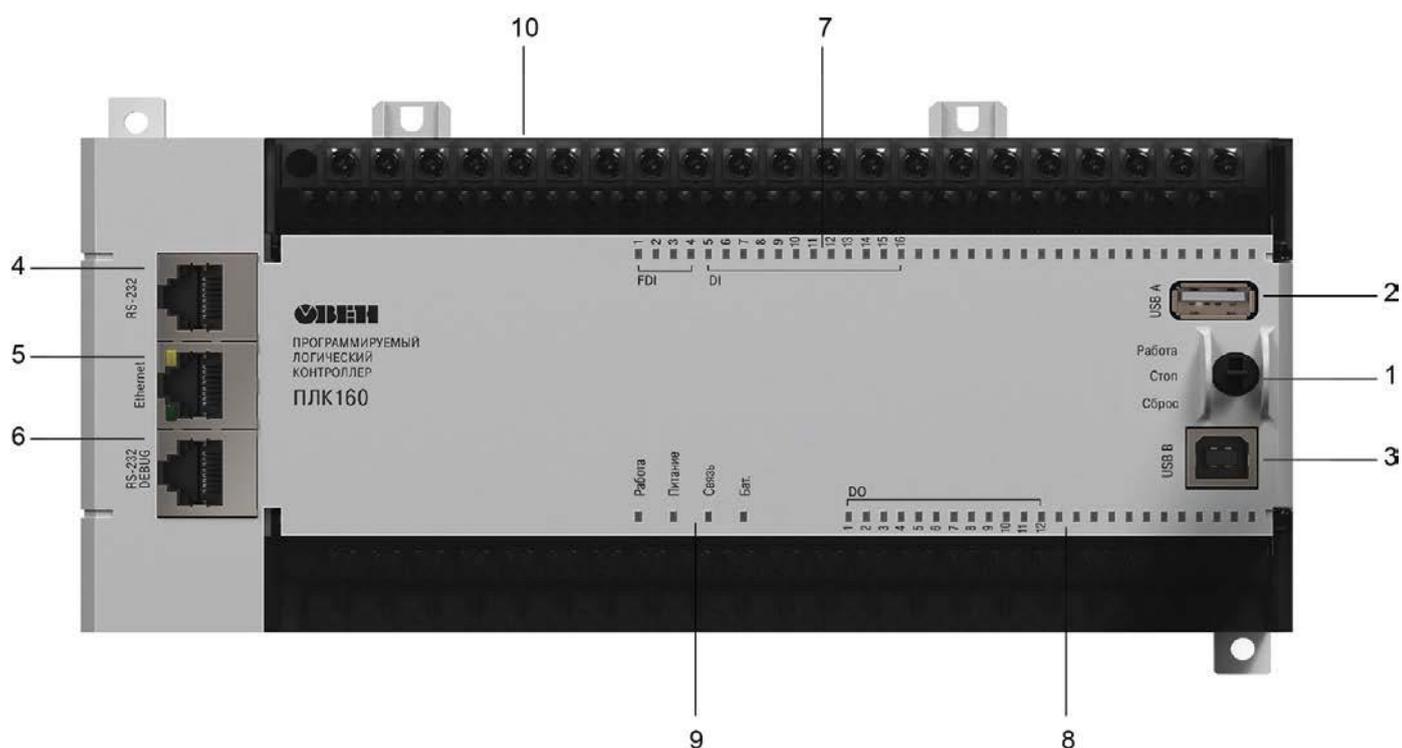


Рис. 3. Внешний вид ОВЕН ПЛК-160

Таблица 1

№ п/п	Элемент	Функциональное назначение
1	Переключатель работа/стоп/сброс	Положения переключателя определяют состояния прибора
2	USB-Host (Разъем USB Type-A)	Порт для подключения USB-flash накопителей
3	USB-Device (Разъем USB Type-B)	Порт для программирования
4	RS-232 Разъем RJ45	Последовательный интерфейс для подключения устройств
5	Ethernet Разъем RJ-45	Порт Ethernet 100 Base-T предназначен для подключения ПЛК в локальные сети, передачи данных и программирования
6	RS-232 Debug (Разъем RJ-45)	Последовательный интерфейс предназначен для подключения устройств и программирования
7	Светодиодные индикаторы состояния входов	FDI – обозначение для быстрых входов контроллера; DI – обычные дискретные входы
8	Светодиодные индикаторы состояния выходов	DO – обычные дискретные выходы
9	Светодиодные индикаторы состояния	Индикация состояния контроллера: «Работа», «Питание», «Связь», «Батарея»
10	Съемные клеммные колодки	Для подключения питания прибора, дискретных датчиков, исполнительных механизмов, аналоговых входов и выходов, интерфейсов RS-485 и клеммы встроенного источника постоянного напряжения 24 В

Описание состояний ПЛК, индицируемых светодиодами «Работа», «Питание», «Связь», «Батарея» представлены в табл. 2 и основные параметры в табл. 3.

Таблица 2

Индикатор	Состояние индикатора	Описание
«Работа»	Слабо светится	Ядро ОС еще не загрузилось после включения питания контроллера
	Мигает раз в 500 мс	Ядро ОС повреждено (не совпадает контрольная сумма)
	Мигает раз в 200 мс	Перегрузка центрального процессора
	Светится	Ядро ОС загружено успешно, программа CODESYS загрузилась и запустилась
	Не светится	Программа CODESYS не работает, остановлена или не загружена
«Питание»	Светится	Наличие питания у контроллера
	Не светится	Отсутствие питания у контроллера
«Связь»	Светится	Наличие связи с CODESYS
	Не светится	Отсутствие связи с CODESYS
«Бат.»	Светится	Необходимо заменить батарейку
	Не светится	Батарейка не требует замены

«FDI», «DI»	Светится	Соответствующий вход замкнут
	Не светится	Вход разомкнут
«DO»	Светится	Соответствующий выход замкнут
	Не светится	Выход разомкнут

Таблица 3

Параметр	Значение
Модель	ПЛК-160
Размеры, Ш x В x Г	208 x 110 x 83 мм
Потребляемая мощность, не более	45 ВА
Напряжение питания	от 90 до 264 В переменного тока (номинальное 120/230 В) частотой от 47 до 63 Гц (номинальное значение 50 Гц)
Встроенные цифровые входы/выходы	16 входов / 12 выходов
Встроенные аналоговые входы/выходы	8 входов / 4 выхода
Центральный процессор	RISC-процессор Texas Instruments Sitara AM1808
Объем оперативной памяти	Пользовательская программа 1 Мбайт, данные пользовательской программы 128 Кбайт, heap до 4 Мбайт в зависимости от использования ресурсов (сокеты, конфигурация и др.) (SDRAM), RAM-диск 8 Мбайт
Объем энергонезависимой памяти (FLASH)	6 Мбайт доступно для хранения файлов и архивов
Размер Retain-памяти (MRAM)	16 Кбайт
Количество сокетов	30
Время выполнения пустого цикла	Установленное по умолчанию (стабилизированное) – 1 мс (настраивается в окне «Конфигурация ПЛК (PLC Configuration) CODESYS». Настоятельно не рекомендуется устанавливать время цикла, равное 0 мс
Интерфейсы связи	RS-485, RS-232, RS-232 Debug, Ethernet 100 Base-T, USB-Device, USB-Host

Ниже представлены схемы подключения входных и выходных цепей ОВЕН ПЛК -160 (рис. 4).

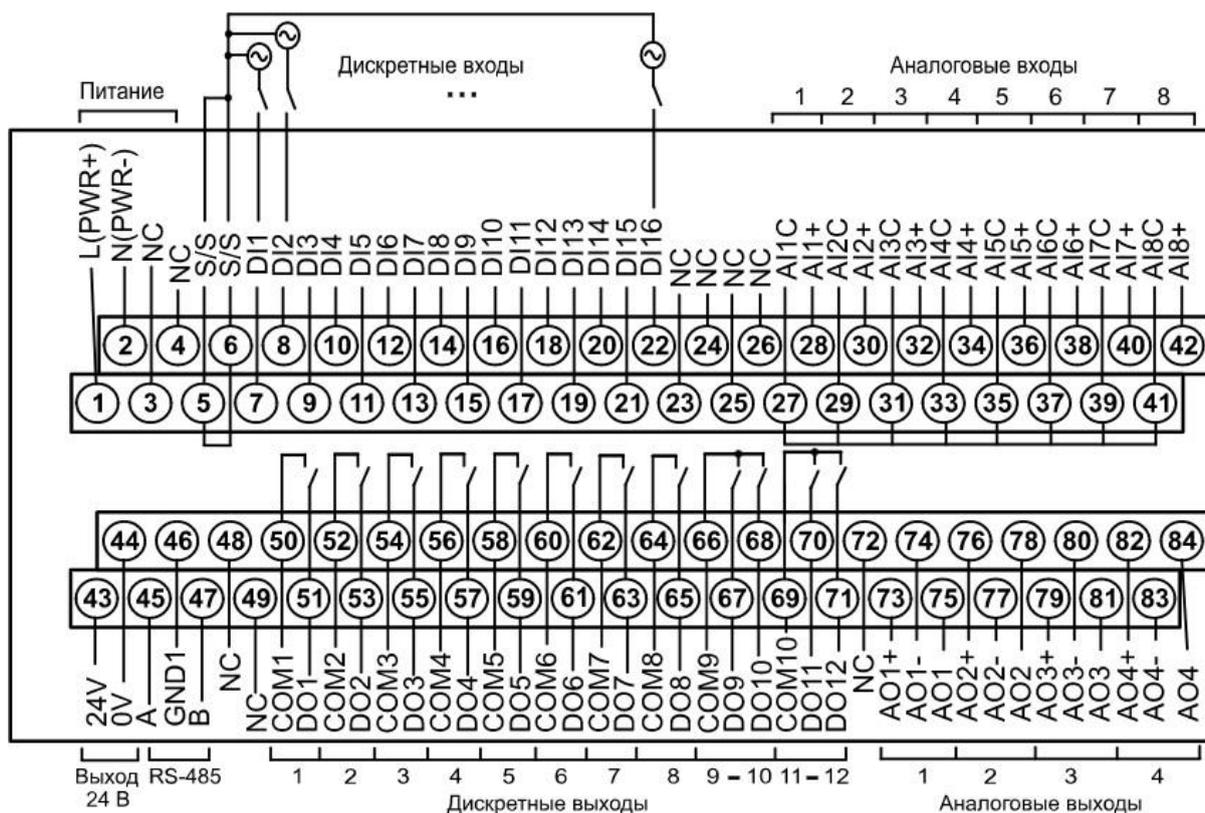


Рис. 4. Схемы подключения входных и выходных цепей контроллера ПЛК-160

Технические характеристики входных и выходных цепей контроллера представлены в табл.4.

Таблица 4

Параметр	Значение
Дискретные входы	
Количество входов из них быстродействующих	16 4 (DI1–DI4)
Тип входов по ГОСТ Р 52931	1 и 2
Напряжение питания дискретных входов	24 ± 3 В
Максимальный входной ток дискретного входа, не более	7 мА при питании 24 В, 8,5 мА при питании 27 В
Сигнал «логической единицы», соответствующий состоянию «Включено», дискретных входов для постоянного напряжения (ток в цепи)	От 15 до 30 В (ток от 3 до 15 мА)
Сигнал «логического нуля», соответствующий состоянию «Выключено», дискретных входов для постоянного напряжения (ток в цепи)	От минус 3 до 5 В (ток до 15 мА)

Минимальная длительность импульса, воспринимаемого дискретным входом:	
для обычных входов	1 мс
для быстродействующих	0,02 мс
Подключаемые входные устройства	Коммутационные устройства (контакты кнопок, выключателей, герконов, реле и т. п.),
	датчики, имеющие на выходе транзистор n-p-n или p-p-n типа с открытым коллектором,
	дискретные сигналы 24 ± 3 В
Аналоговые входы	
Количество аналоговых входов	8
Тип поддерживаемых унифицированных сигналов	Ток от 0 (4) до 20 мА, ток от 0 до 5 мА, напряжение от 0 до 10 В
Разрядность АЦП	14 бит
Входное сопротивление, не более:	
в режиме измерения тока	170 Ом
в режиме измерения напряжения, не менее	200 кОм
Период опроса одного входа	10 мс
Предел основной приведенной погрешности преобразования	$\pm 0,25$ %
Предел дополнительной приведенной погрешности преобразования на каждые 10 градусов изменения температуры	$\pm 0,05$ %
Дискретные выходы	
Количество релейных выходных каналов	12
Максимальный ток, коммутируемый контактами реле, не более	3 А (для переменного напряжения не более 250 В, частотой 50 Гц и $\cos \varphi > 0,4$ – нагрузка для категории использования АС-15 по ГОСТ IEC 60947-1), 3 А (для постоянного напряжения не более 30 В – нагрузка для категории использования DC-13 по ГОСТ IEC 60947-1)
Время переключения контактов реле из состояния «лог. 0» в «лог. 1» и обратно, не более	50 мс (выходы DO1–DO12)
Механический ресурс реле, не менее	300 000 циклов переключений при максимальной коммутируемой нагрузке, 500 000 циклов переключений при коммутации нагрузки менее половины от максимальной

Аналоговые выходы			
Количество аналоговых выходов	4		
Тип выходного сигнала	Универсальный, ток от 4 до 20 мА, напряжение от 0 до 10 В	Напряжение от 0 до 10 В	Ток от 4 до 20 мА
Сопротивление нагрузки	Не более 500 Ом для 4...20 мА, не менее 2000 Ом для 0...10 В	Не менее 2000 Ом	Не более 500 Ом
Предел основной приведенной погрешности ЦАП	± 0,5 %		
Разрядность ЦАП	12 бит	10 бит	10 бит
Минимальный период обновления выходов	100 мс		
Питание аналоговых выходов, внешнее	24 ± 3 В, длина линии от источника питания не должна превышать 30 м		
Предел допускаемой дополнительной приведенной погрешности аналоговых выходов, вызванной изменением температуры окружающего воздуха от нормальной на каждые 10 °С изменения температуры	Не более 0,5 предела допускаемой основной приведенной погрешности		

Для более подробного изучения технических характеристик следует обратиться к технической документации на изучаемый программируемый логический контроллер.

Для имитации пульта оператора используется блок дискретного ввода, который содержит 8 двухпозиционных переключателей DI1 – DI8, подсоединенных ко входам ПЛК DI1 – DI8 соответственно.

Индикация включенного состояния выходных цепей ПЛК осуществляется за счет блока дискретного вывода, который содержит 8 светодиодов DO1 – DO8, подсоединенных к выходам ПЛК DO1 – DO8 соответственно.

Для включения питания модуля используется кнопка с фиксацией и подсветкой.

Основы программирования ПЛК

Для подготовки программ при проведении лабораторных работ служит среда программного обеспечения *CoDeSys*. Для запуска программы необходимо на рабочем столе *Windows* дважды щелкнуть курсором мыши по соответствующему ярлыку.

После этого откроется пустое окно программы *CoDeSys* (рис. 5), в котором можно создать новый проект или открыть уже созданный проект.

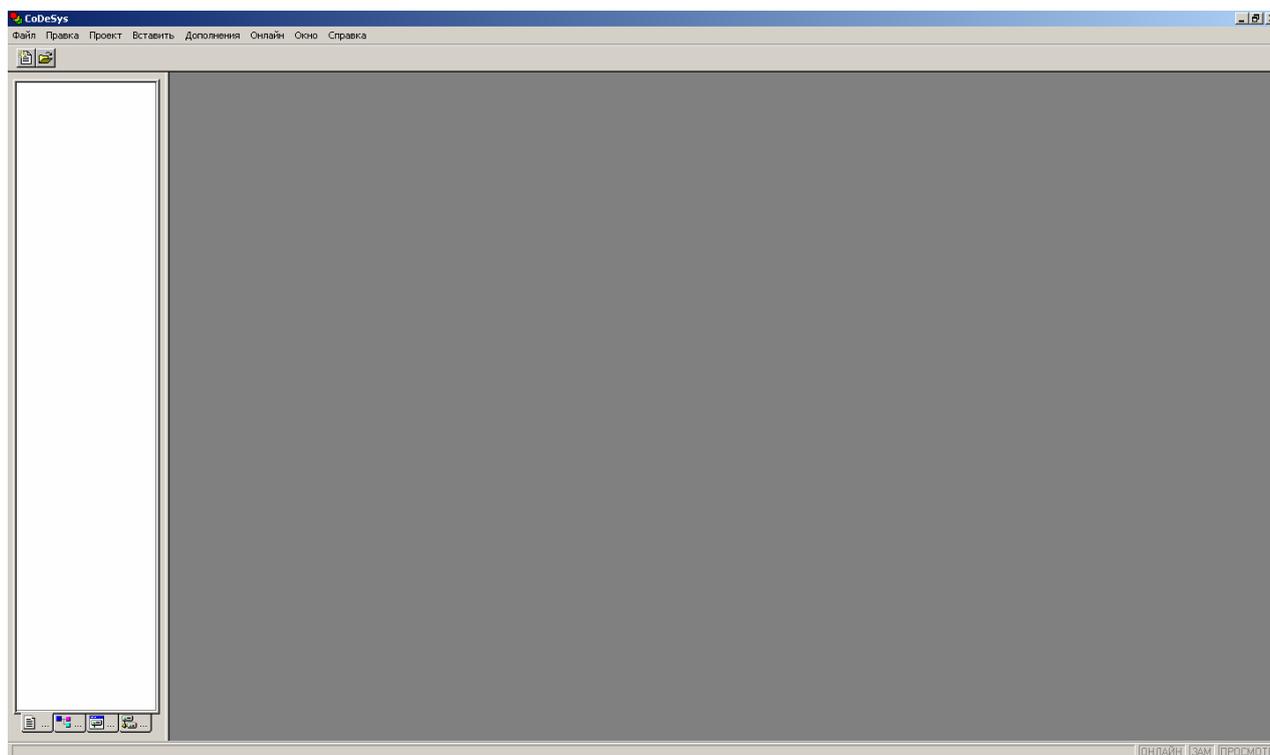


Рис. 5. Стартовое окно *CoDeSys*

Для создания нового проекта необходимо в выпадающем меню «Файл» выбрать пункт «Создать», после чего появится окно конфигурирования проекта «Настройка целевой платформы» (рис. 6). В выпадающем списке вариантов конфигурации (*target*-файлов) выбрать используемый в стенде контроллер ПЛК-160. *Target*-файл содержит информацию о ресурсах контроллера: количестве и типах входов и выходов, интерфейсов, памяти, дополнительных устройствах и т.д., с которыми работает программная среда *CoDeSys*. После выбора контроллера появится окно общих настроек выбранного ПЛК (рис. 7), в котором необходимо выставить те же параметры, которые представлены на рисунке. После завершения настройки необходимо нажать «ОК».

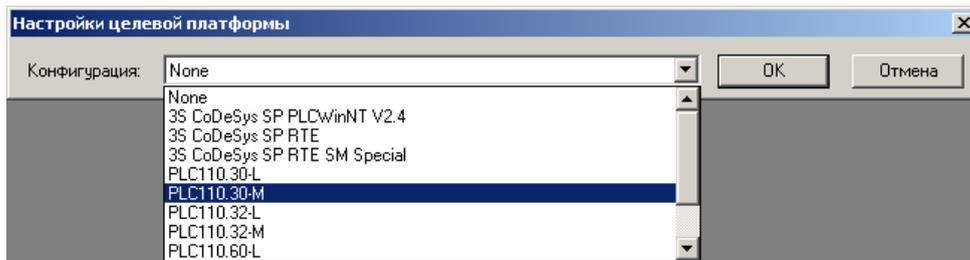


Рис. 6. Окно выбора контроллера

После конфигурирования проекта откроется окно настройки программных компонентов (POU) проекта (рис. 8), в котором выбирается имя и тип POU, а также язык реализации управляющей программы. После завершения настройки также необходимо нажать «ОК».

CoDeSys поддерживает следующие программные компоненты (POU):

- 1) программный блок (PRG);
- 2) функциональный блок (FB);
- 3) функция (FC).

CoDeSys поддерживает следующие языки программирования:

- 1) текстовые: список инструкций (IL) и структурированный текст (ST);
- 2) графические: язык последовательных функциональных схем (SFC), язык функциональных блоковых диаграмм (FBD), язык релейных диаграмм (LD)

Кроме того, CoDeSys включает поддержку основанного на языке FBD редактора непрерывных функциональных схем (CFC).

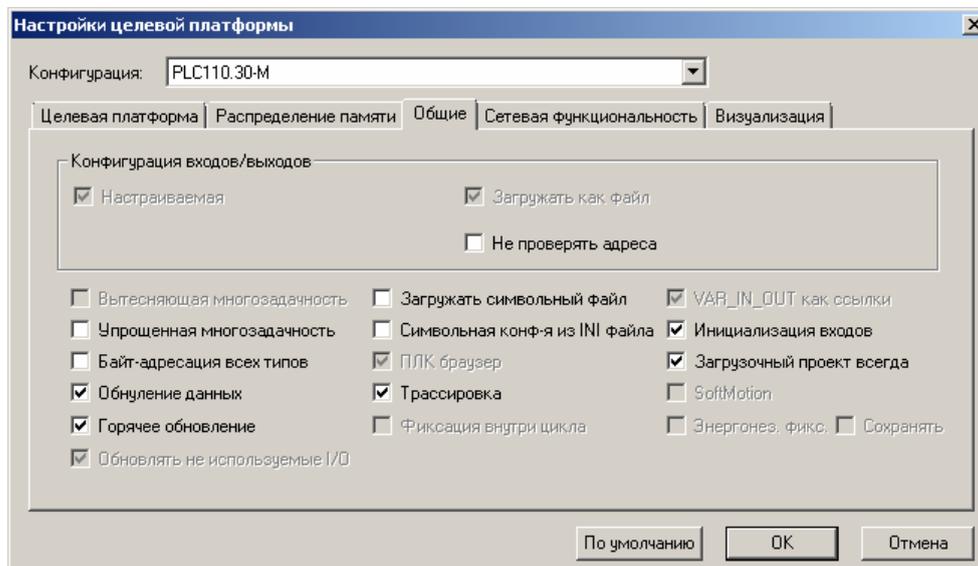


Рис. 7. Окно настройки параметров контроллера

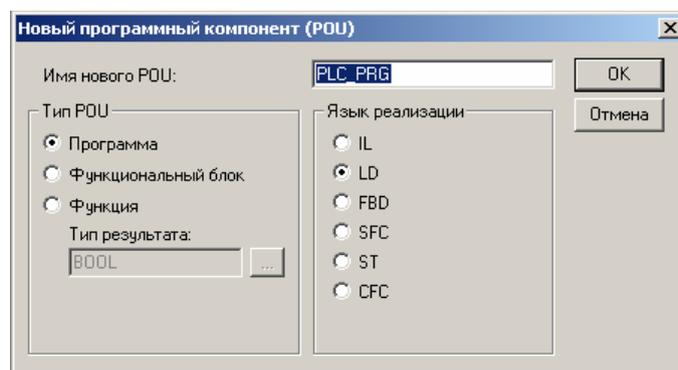


Рис. 8. Окно настройки *POU* проекта

В CoDeSys введено несколько языков для того, чтобы пользователь мог применить язык, наиболее удобный для него: программисты выбирают язык Π или ST, специалисты, имеющие опыт работы с релейной логикой, выбирают язык LD, специалисты по системам автоматического управления (САУ) и схемотехнике выбирают привычный для них язык FBD.

Выбор одного из шести языков определяется не только предпочтениями пользователя, но и смыслом решаемой задачи:

Если исходная задача формулируется в терминах последовательной обработки и передачи сигналов, то для нее проще и нагляднее использовать язык FBD. Если задача описывается как последовательность срабатываний контактов и реле, то для нее нагляднее всего будет язык LD. Для задач, которые изначально формулируются в виде сложного разветвленного алгоритма, удобнее будет язык ST.

В результате конфигурирования откроется рабочее окно программы (рис. 9).

Всю рабочую область программы можно условно разделить на несколько частей:

- I – окно менеджера проекта;
- II – область объявления переменных проекта;
- III – рабочая область;
- IV – панели инструментов;
- V – окно сообщений;
- VI – строка состояния.

В менеджере проекта все настройки проекта распределены по четырем закладкам:

1) Менеджер программных компонентов (*POU*). В данной закладке можно редактировать текущие или добавлять новые компоненты (программы, блоки, функции).

2) Менеджер данных. В данном окне есть возможность добавлять и редактировать уже созданные базы данных проекта.

3) Менеджер визуализаций. В этой закладке есть возможность управления (создание и редактирование свойств) экранов для системы визуализации.

4) Менеджер ресурсов ПЛК объединяет ряд меню, необходимых для конфигурирования свойств ПЛК: различные библиотеки, конфигураторы контроллера, задач и тревог, менеджеры библиотек и параметров и ряд других.

В рабочей области проекта производится непосредственно набор управляющей программы.

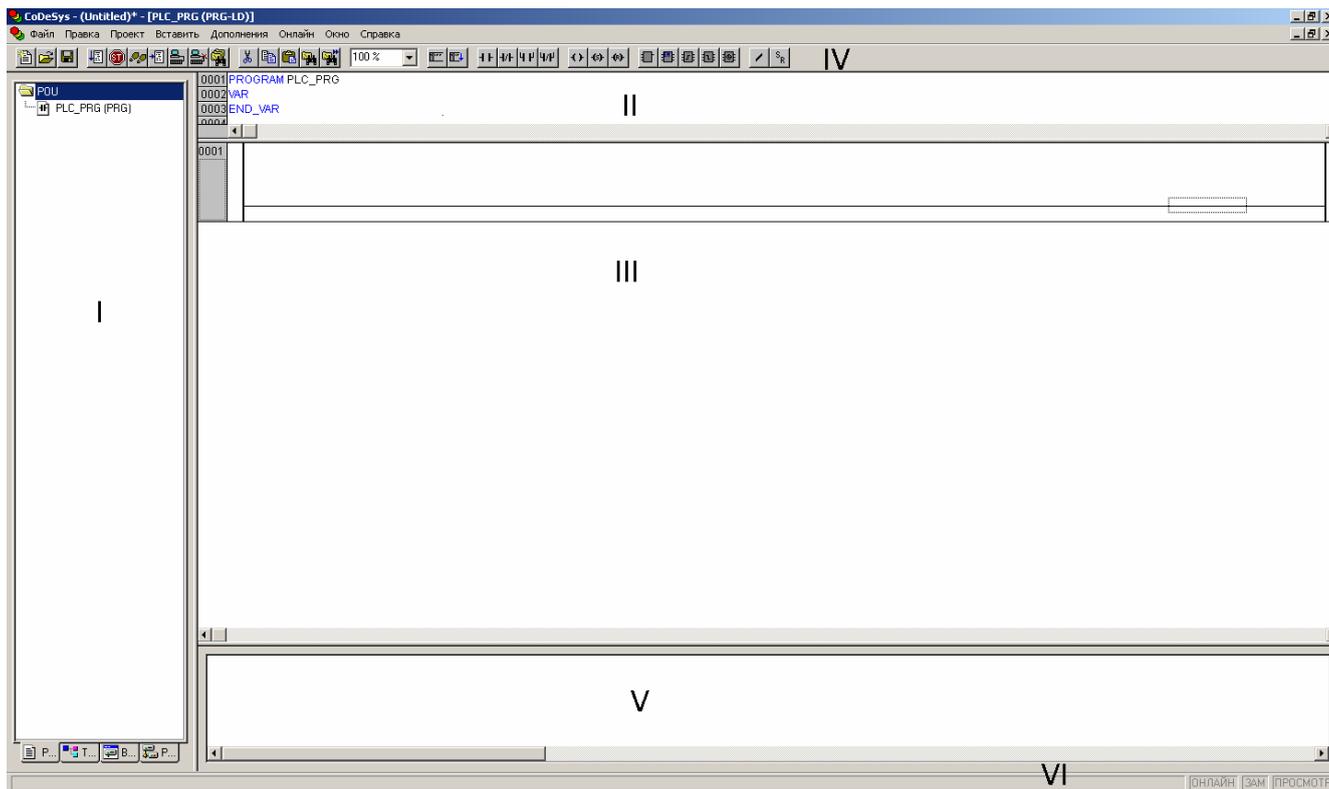


Рис. 9. Рабочее окно программы *CoDeSys 2.3*

Панель инструментов (рис. 10) используется для создания пользовательской управляющей программы, также компиляции, отладки и загрузки проекта в память ПЛК.



Рис. 10. Панель инструментов

При выборе в окне настроек различных языков программирования, например *LD*, *FBD* и *SFC*, панель инструментов принимает различный вид. В табл. 5 представлено краткое описание функций, доступ к которым осуществляется с панели инструментов при программировании на языке *LD*, а также представлены пиктограммы панели инструментов для других языков МЭК: *FBD* и *SFC*.

Таблица 5

Язык	№ п/п	Пиктограмма	Описание
LD	1		Вставка нового сегмента до текущего сегмента
	2		Вставка нового сегмента после текущего сегмента
	3		Вставка замыкающего контакта
	4		Вставка размыкающего контакта

Язык	№ п/п	Пиктограмма	Описание	
	5		Вставка параллельного замыкающего контакта	
	6		Вставка параллельного размыкающего контакта	
	7		Вставка выходной катушки	
	8		Установка состояния <i>SET</i> обмотки выходного реле	
	9		Установка состояния <i>RESET</i> обмотки выходного реле	
	10		Установка функционального блока	
	11		Установка функционального блока со входом <i>EN</i>	
	12		Установка детектора переднего фронта	
	13		Установка детектора заднего фронта	
	14		Установка таймера <i>TON</i>	
	15		Установка инверсии	
	16		Установка функции <i>SET/RESET</i> на обмотке выходного реле	
	FBD	1		Добавление входа в функциональный блок
		2		Добавление выхода из функционального блока
		3		Добавление элемента схемы (функционального блока)
		4		Присваивание
5			Переход	
6			Возврат	
7			Инверсия	
8			<i>SET/RESET</i>	
SFC	1		Шаг-переход (сверху)	
	2		Шаг-переход (снизу)	
	3		Альтернативная ветвь (справа)	
	4		Альтернативная ветвь (слева)	
	5		Параллельная ветвь (справа)	
	6		Параллельная ветвь (слева)	
	7		Безусловный переход	
	8		Переход – безусловный переход	
	9		Использовать МЭК-шаги	

Далее в соответствии с поставленной задачей необходимо разработать алгоритм автоматизации управления объектом либо в виде логических уравнений, либо в виде схемы алгоритма.

Разработчик присваивает входным, выходным сигналам и внутренним переменным адреса в соответствии с конфигурацией контроллера.

Адресация контроллера имеет следующие особенности:

- прямое указание адреса дает способ непосредственного обращения к конкретной области памяти;

- использование области памяти через объявленную особым образом переменную (входную, выходную или локальную).

Прямой адрес образуется из префикса «%», префиксов области памяти и размера, одного или нескольких целых чисел, разделенных точкой.

Префиксы области памяти: *I* – входы, *Q* – выходы, *M* – память данных.

Префиксы размера: *X* – один бит, *B* – байт (8 бит), *W* – слово (16 бит), *D* – двойное слово (32 бит).

Примеры:

- 1) %*QX*7.5 и %*Q*7.5 – бит 7.5 в области выходов;
- 2) %*IW*215 – 215-е слово в области входов;
- 3) %*QB*7 – байт 7 в области выходов;
- 4) %*MD*48 – двойное слово в позиции памяти 48;
- 5) %*IW*2.5.7.1 – зависит от конфигурации ПЛК;

По умолчанию конфигурация ПЛК имеет вид, представленный в табл. 6.

Таблица 6

Входные сигналы				Выходные сигналы			
Адрес ПЛК	Адрес пользователя						
% <i>IX</i> 0.0	<i>I</i> 1	% <i>IX</i> 1.0.6	<i>I</i> 9	% <i>QX</i> 2.0	<i>Q</i> 1	% <i>QX</i> 3.0.2	<i>Q</i> 7
% <i>IX</i> 0.1	<i>I</i> 2	% <i>IX</i> 1.0.7	<i>I</i> 10	% <i>QX</i> 2.1	<i>Q</i> 2	% <i>QX</i> 3.0.3	<i>Q</i> 8
% <i>IX</i> 1.0.0	<i>I</i> 3	% <i>IX</i> 1.1.0	<i>I</i> 11	% <i>QX</i> 2.2	<i>Q</i> 3	% <i>QX</i> 3.0.4	<i>Q</i> 9
% <i>IX</i> 1.0.1	<i>I</i> 4	% <i>IX</i> 1.1.1	<i>I</i> 12	% <i>QX</i> 2.3	<i>Q</i> 4	% <i>QX</i> 3.0.5	<i>Q</i> 10
% <i>IX</i> 1.0.2	<i>I</i> 5	% <i>IX</i> 1.1.2	<i>I</i> 13	% <i>QX</i> 3.0.0	<i>Q</i> 5	% <i>QX</i> 3.0.6	<i>Q</i> 11
% <i>IX</i> 1.0.3	<i>I</i> 6	% <i>IX</i> 1.1.3	<i>I</i> 14	% <i>QX</i> 3.0.1	<i>Q</i> 6	% <i>QX</i> 3.0.7	<i>Q</i> 12
% <i>IX</i> 1.0.4	<i>I</i> 7	% <i>IX</i> 1.1.4	<i>I</i> 15				
% <i>IX</i> 1.0.5	<i>I</i> 8	% <i>IX</i> 1.1.5	<i>I</i> 16				

Ниже рассматривается пример программирования на языке *LD*.

При программировании на языке *LD* в виде релейно-контактной схемы программа разделяется на сегменты. Каждый сегмент представляет собой отдельную цепь, по которой может протекать ток. Шина питания находится слева (вертикальная линия). Таким образом, каждому логическому уравнению соответствует свой сегмент.

По умолчанию в блоке *PLC_PRG (PRG)* уже есть пустой шаблон сегмента. Для создания нового сегмента необходимо на панели инструментов нажать пиктограмму

 «Цепь (Перед)» или  «Цепь (После)»

Рассмотрим ввод программы на языке лестничных диаграмм (*LD*) для реализации простого уравнения:

$$Q0 = I1 \cdot \bar{I2} + I3$$

В адресах контроллера это уравнение примет следующий вид:

$$\%QX\ 2.0 = \%IX\ 0.0 \cdot \%IX\ 0.1 + \%IX\ 1.0.0$$

- 1) Установить в токовую цепь (сегмент) обмотку (рис. 11а). Для этого нажать на кнопку .

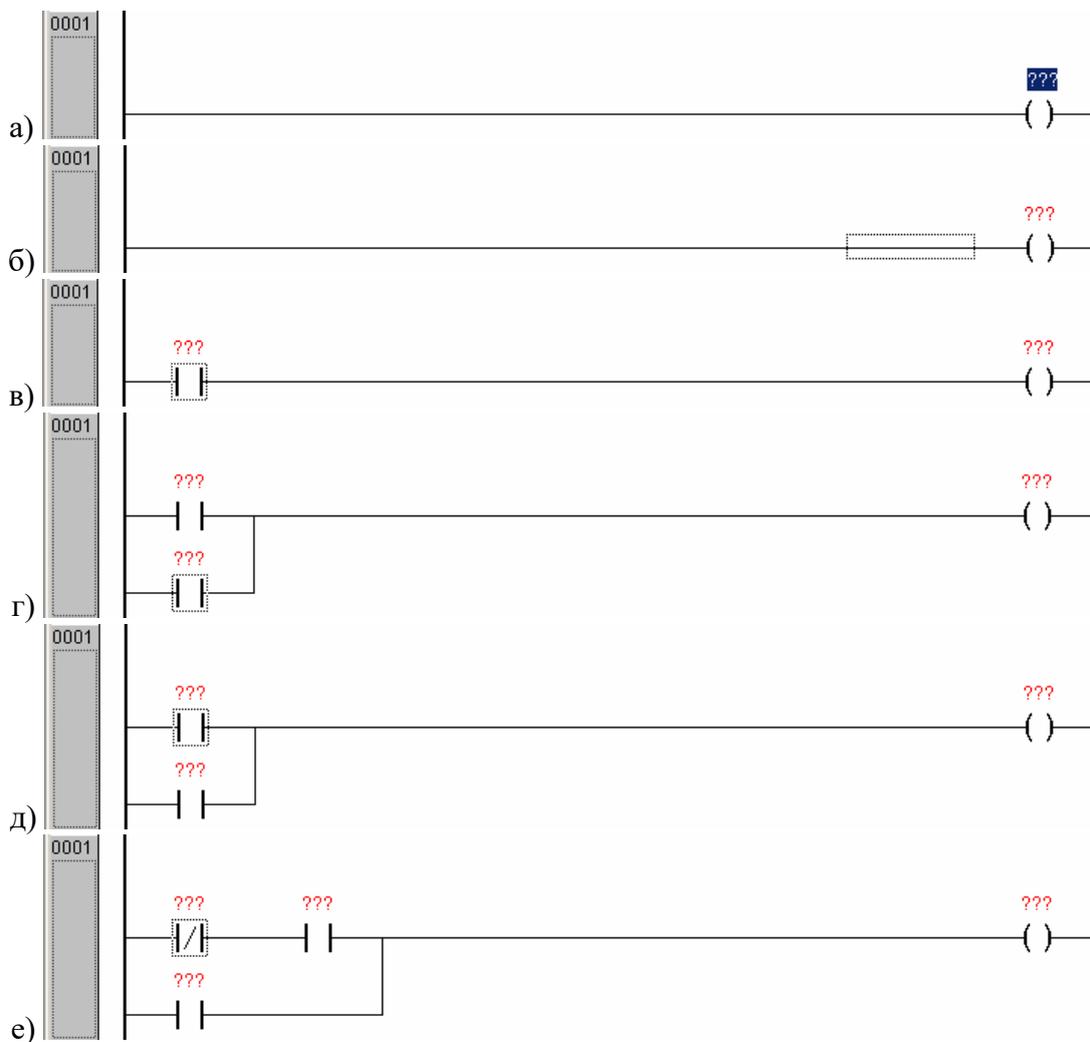


Рис. 11.

- 2) Щёлкнуть по тому месту (непосредственно по линии), куда нужно установить замыкающий (нормально открытый) контакт (рис. 11б).
- 3) Вставить замыкающий контакт, нажав на пиктограмму  (рис. 11в).
- 4) Вставить параллельный замыкающий контакт, нажав на пиктограмму  (рис. 11г).
- 5) Выделить место, куда необходимо поставить нормально замкнутый (размыкающий) контакт (рис. 11д).
- 6) Вставить размыкающий контакт, нажав на пиктограмму  (рис. 11е).
- 7) Теперь необходимо ввести адреса контактов. Для этого щёлкнуть по знакам ??? над контактом и ввести адрес (ввод адреса завершается нажатием клавиши «ENTER» рис. 12).



Рис. 12.

Для удобства пользователя при разработке программы доступна символьная адресация. Для этого предварительно необходимо в окне задания переменных проекта объявить переменные.

Для этого необходимо выбрать элемент на схеме, ввести имя переменной и нажать «ENTER». В открывшемся окне необходимо выбрать класс, тип, адрес переменной и нажать «OK» (рис. 13). После объявления всех входных и выходных переменных программа примет следующий вид (рис. 14), а окно задания переменных – рис. 15.

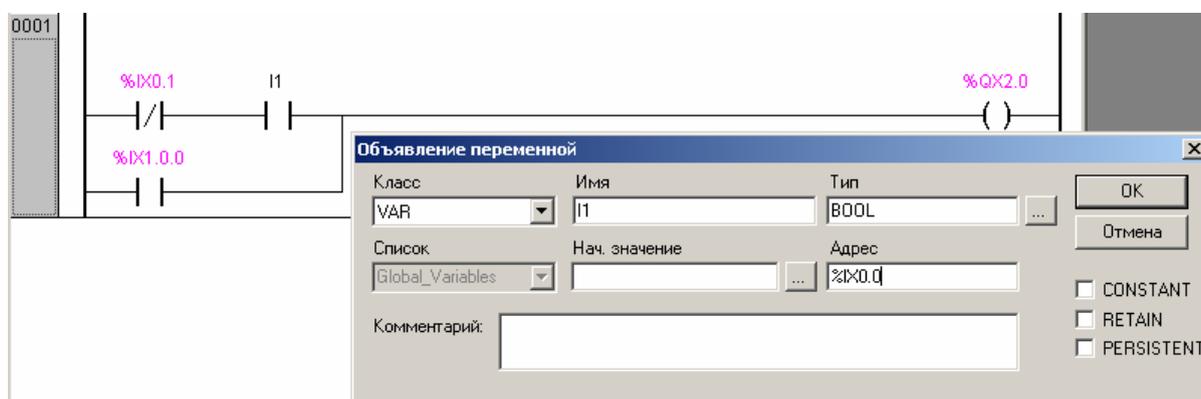


Рис. 13.



Рис. 14.

```

PLC_PRG (PRG-LD)
0001 PROGRAM PLC_PRG
0002 VAR
0003   I1 AT %IX0.0: BOOL;
0004   I2 AT %IX0.1: BOOL;
0005   I3 AT %IX1.0.0: BOOL;
0006   Q1 AT %QX2.0: BOOL;
0007 END_VAR

```

Рис. 15.

Для удаления какого-либо элемента программы необходимо его выделить и нажать на клавиатуре клавишу *Delete*. Для отмены предыдущего действия необходимо выбрать в меню «Правка» пункт «Отменить».

В случае некорректного ввода адреса или символического имени программа высвечивает некорректное обозначение красным цветом.

После ввода программы необходимо скомпилировать ее. Для этого в выпадающем меню «Проект» необходимо выбрать пункт «Компилировать», либо нажать «быструю» клавишу *F11*. Лог компиляции программы будет выведен в окно сообщений. Если программа набрана правильно (рис. 16а), то последняя строка будет вида: 0 ошибок, 0 предупреждений. В том случае, если в программе будут найдены ошибки (рис. 16б), то они будут выделены в логе красным цветом. Также в последней строке будет: *X* ошибок, *X* предупреждений. После исправления ошибок необходимо еще раз откомпилировать программу. После компиляции программа готова к загрузке в память ПЛК.

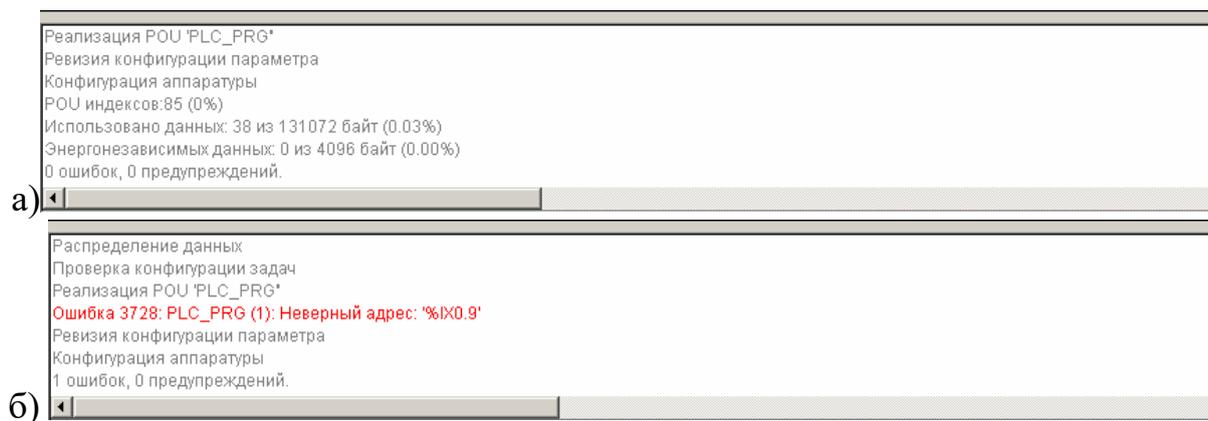


Рис. 16. Примеры сообщений при проведении компиляции пользовательской программы

Далее необходимо рассмотреть реализацию этого же логического уравнения ($Q0 = I1 \cdot \overline{I2} + I3$) на других языках:

1) **на языке инструкций (IL)** программа будет иметь достаточно простой и понятный вид:

LDN %IX1.0.0 (загрузить инверсный контакт)

AND %IX0.0 (загрузить контакт)

OR %IX0.1 (загрузить параллельный контакт)

ST %QX2.0 (результат выдать на обмотку)

2) разработка программы **на языке функциональных блоковых диаграмм (FBD)** ведется в следующей последовательности:

– в свойствах проекта настроить язык программирования *FBD*;

– на панели инструментов левой клавишей мыши нажать пиктограмму . На рабочем поле в новом сегменте должно появиться изображение элемента (рис. 17а), В поле названия элемента указать тип элемента *AND (И)*;

– около входов элемента указать адреса %IX0.0 и %IX0.1 сверху вниз соответственно;

– на входе, соответствующем адресу %IX0.1, нажать правую клавишу мыши и выбрать пункт «Инверсия» (рис. 17б);

- установить курсор на выходной части элемента И;
- левой клавишей мыши нажать пиктограмму . В поле названия элемента указать тип элемента *OR* (ИЛИ), напротив свободного входа указать адрес $\%IX1.0.0$ (рис. 17в);
- установить курсор на выходной части элемента *OR* (ИЛИ);
- левой клавишей мыши нажать пиктограмму . В сформированном выходе необходимо указать адрес $\%QX2.0$. Программа примет итоговый вид (рис. 17г).

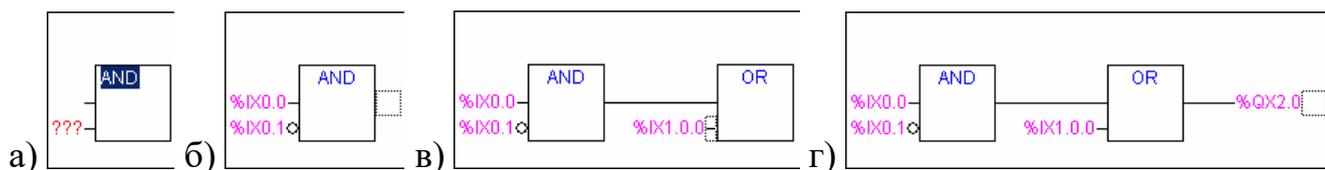


Рис. 17. Программа на языке *FBD*

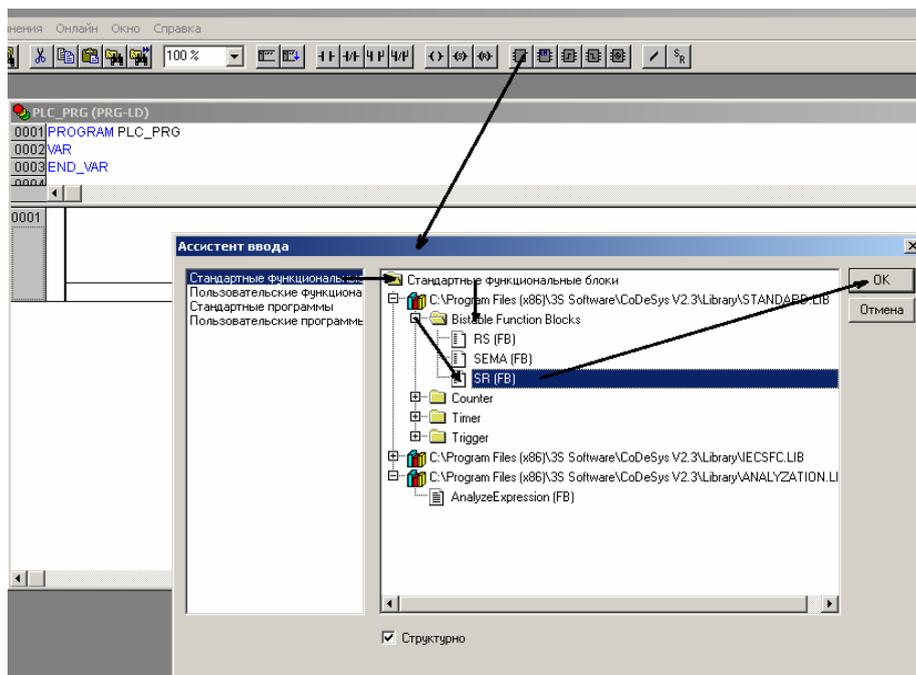
Далее рассматривается вставка функции на примере создания следующей программы: если кратковременно включить тумблер *I1* (адрес $IX0.0$), то включится выход *Q1* (адрес $QX2.0$), если кратковременно включить тумблер *I2* (адрес $IX0.1$), то выход *Q1* выключится. Для этого воспользуемся функцией *SR*-триггера:

1) установить выходную обмотку с адресом $\%QX2.0$, установить контакт с адресом $\%IX0.0$;

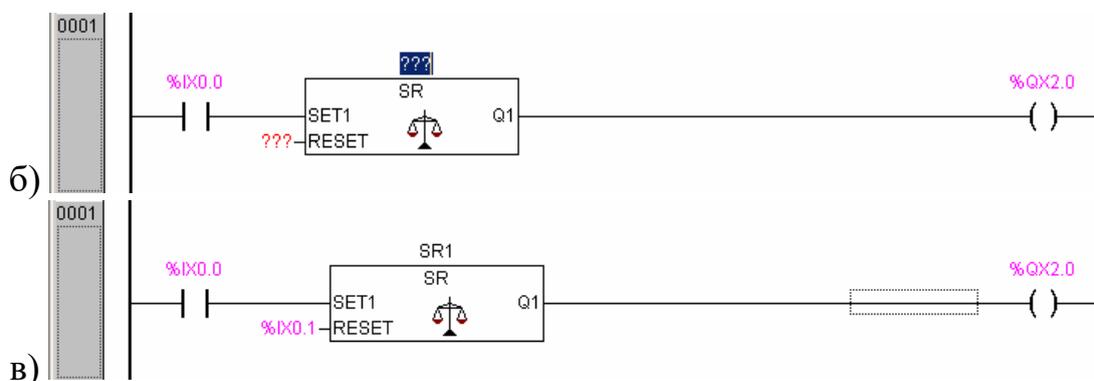
2) установить курсор в новом сегменте. Левой клавишей мыши на панели инструментов нажать пиктограмму . В появившемся окне (рис. 18а) в закладке «Стандартные функциональные блоки» раскрыть библиотеку «*STANDARD*». В открывшемся перечне раскрыть закладку «*Bistable Function Blocks*», в которой выбрать функциональный блок *SR*, после чего нажать «ОК». В сегменте установится блок *SR*-триггера (рис. 18б);

3) в поле ??? над блоком вписать название блока *SR1*;

4) во входной цепи *RESET* на месте ??? установить адрес $\%IX0.1$. Программа примет следующий вид (рис. 18в).



а)



б)

в)

Рис. 18. Пример использования функциональных блоков в лестничных диаграммах

3. Для программирования контроллера ПЛК160 доступно 4 вида таймеров:

- таймер-формирователь импульса (*TP*);
- таймер с задержкой включения (*TON*);
- таймер с задержкой выключения (*TOF*);
- часы реального времени (*RTC*).

В табл. 7 представлены примеры реализации таймеров на языках *IL* и *ST*.

На рис. 19 представлено графическое изображение блоков таймеров, используемое в языке *FBD*. Блок, реализующий таймер, имеет следующие входы и выходы: *IN* – вход запуска, *EN* – вход разрешения, *PT* – уставка (заданное время), *Q* – выход (состояние таймера), *ET* – оставшееся время.

Для задания уставки времени используется идентификатор *T#*. Уставку времени можно задать в *B**C**D*-коде с использованием следующего синтаксиса:

*T#AA**d**BB**h**CC**m**DD**s**EEE**ms*,

где *AA* = сутки, *BB* = часы, *CC* = минуты, *DD* = секунды и *EEE* = миллисекунды.

	Пример объявления	Пример IL	Пример ST
TP	<i>TPInst: TP</i>	<i>CAL TPInst(IN := VarBOOL1, PT := T#5s) LD TPInst.Q ST VarBOOL2</i>	<i>TPInst(IN := VarBOOL1, PT:= T#5s); VarBOOL2 :=TPInst.Q</i>
TON	<i>TONInst: TON</i>	<i>CAL TONInst(IN := VarBOOL1, PT := T#5s) LD TONInst.Q ST VarBOOL2</i>	<i>TONInst(IN := VarBOOL1, PT:= T#5s)</i>
TOF	<i>TOFInst: TOF</i>	<i>CAL TOFInst(IN := VarBOOL1, PT := T#5s) LD TOFInst.Q ST VarBOOL2</i>	<i>TOFInst(IN := VarBOOL1, PT:= T#5s); VarBOOL2 :=TOFInst.Q</i>

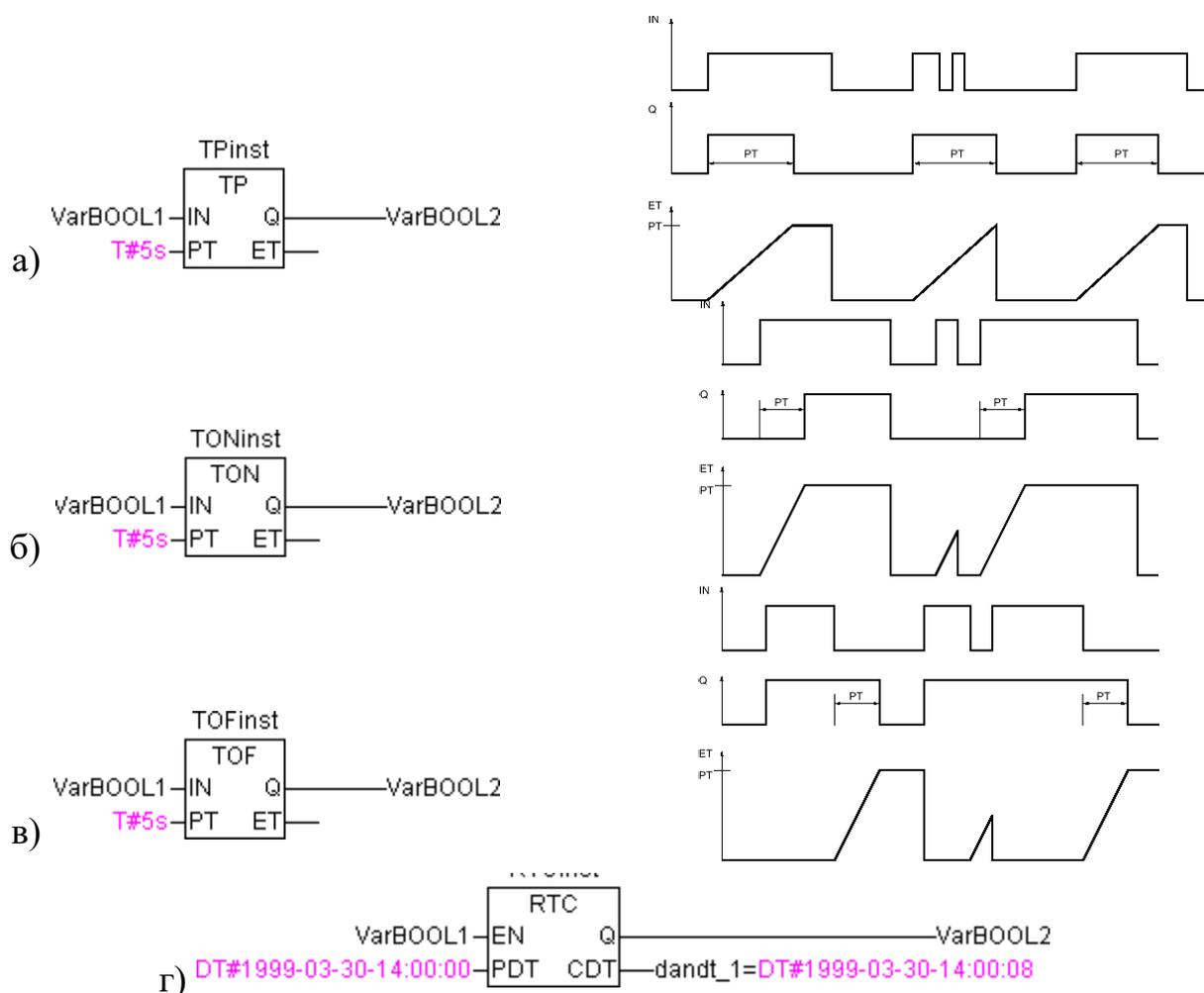


Рис. 19. Таймеры и их временные диаграммы работы

Каждый таймер использует структуру, хранящуюся в блоке данных, для сохранения данных о времени. При этом назначается блок данных, когда устанавливается команда таймера.

В режиме RUN текущее значение таймера уменьшается на одну единицу через интервал времени, установленный базой времени, до тех пор, пока значение времени не станет равным нулю, что соответствует срабатыванию таймера.

4. Для программирования контроллера ПЛК160 доступно 3 вида счетчиков:

- инкрементный (на сложение) счетчик (*CTU*);
- декрементный (на вычитание) счетчик (*CTD*);
- инкрементный/декрементный (реверсивный) счетчик (*CTUD*).

В табл. 8 представлены примеры реализации счетчиков на языках IL и ST. На рис. 20 представлено графическое изображение блоков счетчиков, используемое в языке *FBD*. Блок, реализующий счетчик, имеет следующие входы и выходы: *CU* – вход на сложение, *CD* – вход на вычитание, *PV* – задание (уставка) счетчика, *RESET* – вход сброса, *LOAD* – вход загрузки задания счетчика, *Q*, *QU*, *QD* – выход (состояние) счетчика, *CV* – текущее значение счетчика в двоичном формате.

Таблица 8

	Пример объявления	Пример IL	Пример ST
<i>CTU</i>	<i>CTUInst:</i> <i>CTU</i>	<i>CAL CTUInst(CU := VarBOOL1, RESET := VarBOOL2, PV := VarINT1)</i> <i>LD CTUInst.Q</i> <i>ST VarBOOL3</i> <i>LD CTUInst.CV</i> <i>ST VarINT2</i>	<i>CTUInst(CU:= VarBOOL1, RESET:=VarBOOL2, PV:= VarINT1);</i> <i>VarBOOL3 := CTUInst.Q;</i> <i>VarINT2 := CTUInst.CV</i>
<i>CTD</i>	<i>CTDInst:</i> <i>CTD</i>	<i>CAL CTDInst(CD := VarBOOL1, LOAD := VarBOOL2, PV := VarINT1)</i> <i>LD CTDInst.Q</i> <i>ST VarBOOL3</i> <i>LD CTDInst.CV</i> <i>ST VarINT2</i>	<i>CTDInst(CD:= VarBOOL1, LOAD:=VarBOOL2, PV:= VarINT1);</i> <i>VarBOOL3 := CTDInst.Q;</i> <i>VarINT2 := CTDInst.CV</i>
<i>CTUD</i>	<i>CTUDInst</i> <i>: CUDT</i>	<i>CAL CTUDInst(CU:=VarBOOL2, RESET:=VarBOOL3, LOAD:=VarBOOL4, PV:=VarINT1)</i> <i>LD CTUDInst.Q</i> <i>ST VarBOOL5</i> <i>LD CTUDInst.QD</i> <i>ST VarBOOL5</i> <i>LD CTUInst.CV</i> <i>ST VarINT2</i>	<i>CTUDInst(CU := VarBOOL1, CU:= VarBOOL2, RESET := VarBOOL3, LOAD:=VarBOOL4, PV:= VarINT1);</i> <i>VarBOOL5 := CTUDInst.QU;</i> <i>VarBOOL6 := CTUDInst.QD;</i> <i>VarINT2 := CTUDInst.CV</i>

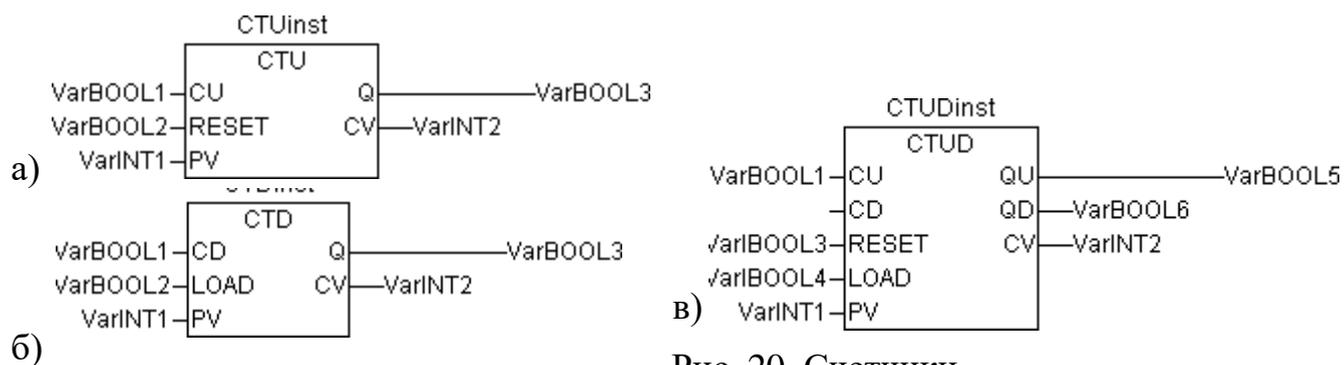


Рис. 20. Счетчики

Уставка счетчика задается либо константой в формате <значение>, либо из области памяти (I, Q, M). Нарастающим фронтом сигнала на входе LOAD в счетчик заносится уставка с входа PV. При нарастающем сигнале на входе CU значение счетчика увеличивается на «1». При нарастающем фронте сигнала на входе CD значение счетчика уменьшается на «1», если значение счетчика больше, чем «0». При нарастающем фронте сигнала на входе RESET счетчик сбрасывается (в счетчик записывается значение «0»).

Для реверсивного счетчика, если оба счетных входа имеют нарастающий фронт, то значение счетчика остается неизменным.

В счетчике CTU выход Q устанавливается в «1», когда значение CV достигает значения PV.

В счетчике CTD выход Q устанавливается в «1», когда значение CV достигает «0».

В счетчике CTUD выход QU устанавливается в «1», когда значение CV больше или равно значению PV, а выход QD – когда значение CV достигает «0».

5. Кроме таймеров и счетчиков в программируемом контроллере ОВЕН ПЛК160 используется стандартный набор операндов и функций, присущий всем контроллерам:

- логические;
- арифметические
- операции сдвига;
- операции сравнения и т.д.

6. Программирование функциональных блоков и функций

Среда разработки CoDeSys предусматривает возможность упростить программирование, если возникает необходимость в программировании управления однотипными технологическими операциями. Ниже рассматриваются, в качестве примера, испытания двигателей бензинового и дизельного. Испытания в принципе одинаковые, только аппараты (кнопки, датчики и устройства индикации) свои. В этом случае необходимо запрограммировать функциональный блок FB, который может многократно вызываться в программе

Например, создадим функциональный блок FB1, которому дадим символическое имя «ENGINE [двигатель]». При кратковременном появлении

команды «turn_on [включить]» двигатель должен включиться и сформировать сигнал «on [включен]».

При достижении фактической скорости «fact [факт]» заданного значения «preset [задание]» формируется сигнал «ready [готов]». Двигатель отключается по сигналу «turn_off [отключить]» или по сигналу «trouble [неисправность]».

Последовательность программирования сводится к следующему:

1) Создать новый проект, в котором создать три программных компонента POU: программный блок PLC_PRG, функциональный блок ENGINE и функция VENT (рис. 21);

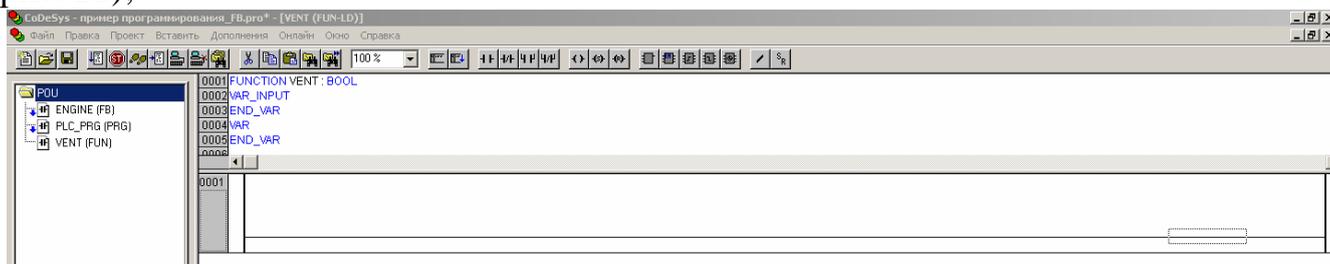


Рис. 21.

2) Объявить переменные для функционального блока ENGINE (рис. 22а):

FUNCTION_BLOCK

ENGINE

VAR_INPUT

turn_on: BOOL;

turn_off: BOOL;

trouble: BOOL;

fact: INT;

preset: INT;

END_VAR

VAR_OUTPUT

on: BOOL;

ready: BOOL;

END_VAR

VAR

END_VAR

3) Объявить переменные для основной программы PLC_PRG (табл. 9 и рис. 22б).

4) Затем разрабатывается программа, управляющая включением и отключением двигателя, как было описано выше.

Таблица 9

Символическое имя	Адрес на модуле	Адрес ПЛК	Тип данных	Комментарии
Входные переменные				
<i>turn_on_B</i>	<i>I1</i>	<i>%IX0.0</i>	<i>BOOL</i>	Включить бензиновый
<i>turn_off_B</i>	<i>I2</i>	<i>%IX0.1</i>	<i>BOOL</i>	Отключить бензиновый
<i>trouble_B</i>	<i>I3</i>	<i>%IX1.0.0</i>	<i>BOOL</i>	Неисправность бензинового

Символическое имя	Адрес на модуле	Адрес ПЛК	Тип данных	Комментарии
<i>turn_on_D</i>	<i>I4</i>	%IX1.0.1	BOOL	Включить дизель
<i>turn_off_D</i>	<i>I5</i>	%IX1.0.2	BOOL	Отключить дизель
<i>trouble_D</i>	<i>I6</i>	%IX1.0.3	BOOL	Неисправность дизеля
Выходные переменные				
<i>on_B</i>	<i>Q1</i>	%QX2.0	BOOL	Бензиновый включен
<i>ready_B</i>	<i>Q2</i>	%QX2.1	BOOL	Заданная скорость достигнута
<i>vent_B</i>	<i>Q3</i>	%QX2.2	BOOL	Вентилятор бензинового включен
<i>on_D</i>	<i>Q5</i>	%QX3.0.0	BOOL	Дизель включен
<i>ready_D</i>	<i>Q6</i>	%QX3.0.1	BOOL	Заданная скорость достигнута
<i>vent_D</i>	<i>Q7</i>	%QX3.0.2	BOOL	Вентилятор дизеля включен
Промежуточные переменные				
<i>fact_B</i>		MW0	INT	Фактическая скорость бензинового двигателя
<i>fact_D</i>		MW2	INT	Фактическая скорость дизеля
Таймеры				
<i>tim_B</i>		TON		Таймер вентилятора бензинового
<i>tim_D</i>		TON		Таймер вентилятора дизеля

```

ENGINE (FB-LD)
0001 FUNCTION_BLOCK ENGINE
0002 VAR_INPUT
0003   turn_on: BOOL;
0004   turn_off: BOOL;
0005   trouble: BOOL;
0006   fact: INT;
0007   preset: INT;
0008 END_VAR
0009 VAR_OUTPUT
0010   on: BOOL;
0011   ready: BOOL;
0012 END_VAR
0013 VAR
0014 END_VAR
0015

```

а)

```

PLC_PRG (PRG-LD)
0001 PROGRAM PLC_PRG
0002 VAR
0003   turn_on_B AT %IX0.0: BOOL;
0004   turn_off_B AT %IX0.1: BOOL;
0005   trouble_B AT %IX1.0.0: BOOL;
0006   turn_on_D AT %IX1.0.1: BOOL;
0007   turn_off_D AT %IX1.0.2: BOOL;
0008   trouble_D AT %IX1.0.3: BOOL;
0009   on_B AT %QX2.0: BOOL;
0010   ready_B AT %QX2.1: BOOL;
0011   vent_B AT %QX2.2: BOOL;
0012   on_D AT %QX3.0.0: BOOL;
0013   ready_D AT %QX3.0.1: BOOL;
0014   vent_D AT %QX3.0.2: BOOL;
0015   tim_B: TON;
0016   tim_D: TON;
0017   fact_B AT %MW0: INT;
0018   fact_D AT %MW2: INT;
0019
0020 END_VAR
0021

```

б)

Рис. 22. Объявление переменных

Набранная программа на языке лестничных диаграмм имеет вид, представленный на рис. 23.

Созданный блок *ENGINE* должен управлять и контролировать работу бензинового и дизельного двигателей. Различные заданные скорости хранятся в отдельных ячейках памяти.

5) Далее необходимо запрограммировать вызов функционального блока в программном блоке. Для этого в сегменте программного блока необходимо установить курсор в заданном положении, на панели инструментов нажать пиктограмму  и установить 2 блока на поле сегмента. В поле названия блока необходимо написать название пользовательского блока (*ENGINE*). Один блок

необходимо назвать «*Gasoline*», другой – «*Diesel*». Соответственно один блок будет отвечать за работу бензинового двигателя, а второй – за работу дизельного (рис. 24а).

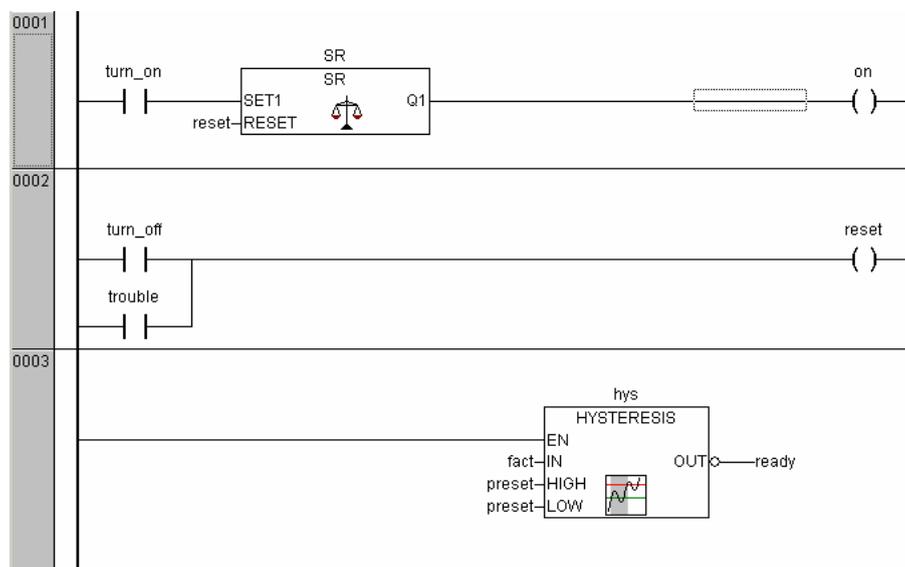


Рис. 23. Функциональный блок *ENGINE*

В поля напротив входов и выходов необходимо установить названия соответствующих переменных. Программный блок после этого примет вид, представленный на рис. 24б. Напротив внутренней переменной «*preset*» блока «*Gasoline*» необходимо установить значение 1500. Таким образом, определена максимальная скорость для бензинового двигателя. Аналогично установить значение 1200 напротив внутренней переменной «*preset*» блока «*Diesel*» (для дизельного двигателя).

б) Далее необходимо сохранить программу;

7) Функции, как и функциональные блоки, расположены в иерархии программы ниже программного блока. Однако в отличие от функционального блока, функции не нужен блок данных. У функций параметры также перечисляются в окне объявления переменных, но статические данные не разрешаются.

В рассматриваемой задаче предусматривается, что каждый двигатель как бензиновый, так и дизель имеет свой вентилятор. Вентилятор включается при работе двигателя и продолжает еще работать 4 с после отключения двигателя. Для управления этими двигателями программируется функция *VENT*.

Необходимо в менеджере проекта дважды щелкнуть на строке *VENT*, чтобы открыть ее. Открывается окно для программирования. Далее нужно объявить переменные:

```
FUNCTION VENT :
  BOOL
  VAR_INPUT
    turn_on: BOOL;
  END_VAR
  VAR
    tim: TON;
  END_VAR
```

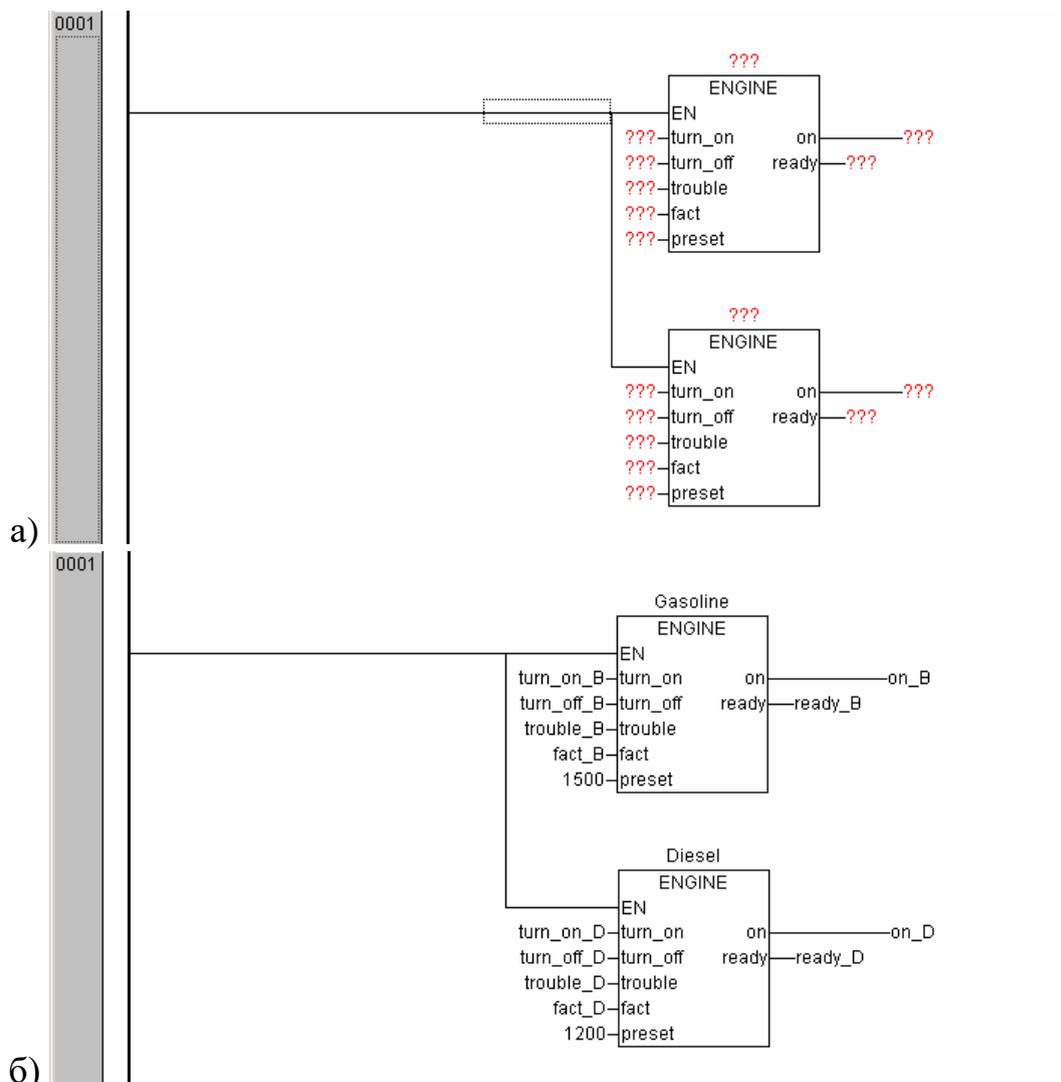


Рис. 24. Сегмент программы управления испытаниями двигателей

8) Затем необходимо запрограммировать управление вентилятором. Набранная программа на языке *FBD* имеет вид, представленный на рис. 25.

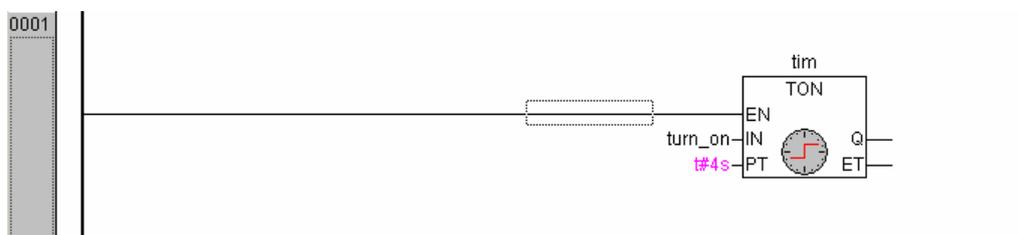


Рис. 25. Программирование функции *VENT*

9) Вызов функции *VENT* в программном блоке выполняется аналогично вызову функционального блока. Все переменные функции снабжаются в программном блоке соответствующими адресами бензинового или дизельного двигателя, которые уже отражены в табл. 9.

В программном блоке *PLC_PRG* вставить новый сегмент. Затем в этом сегменте необходимо установить курсор, на панели инструментов нажать пиктограмму  и

установить 2 блока на поле сегмента. В поле названия блока необходимо списать название пользовательской функции (*VENT*). Одна функция будет отвечать за работу вентилятора бензинового двигателя, а вторая – за работу вентилятора дизельного двигателя (рис. 26а). Далее в соответствующих полях ввести символьные имена из табл. 9. Сохранить программу.

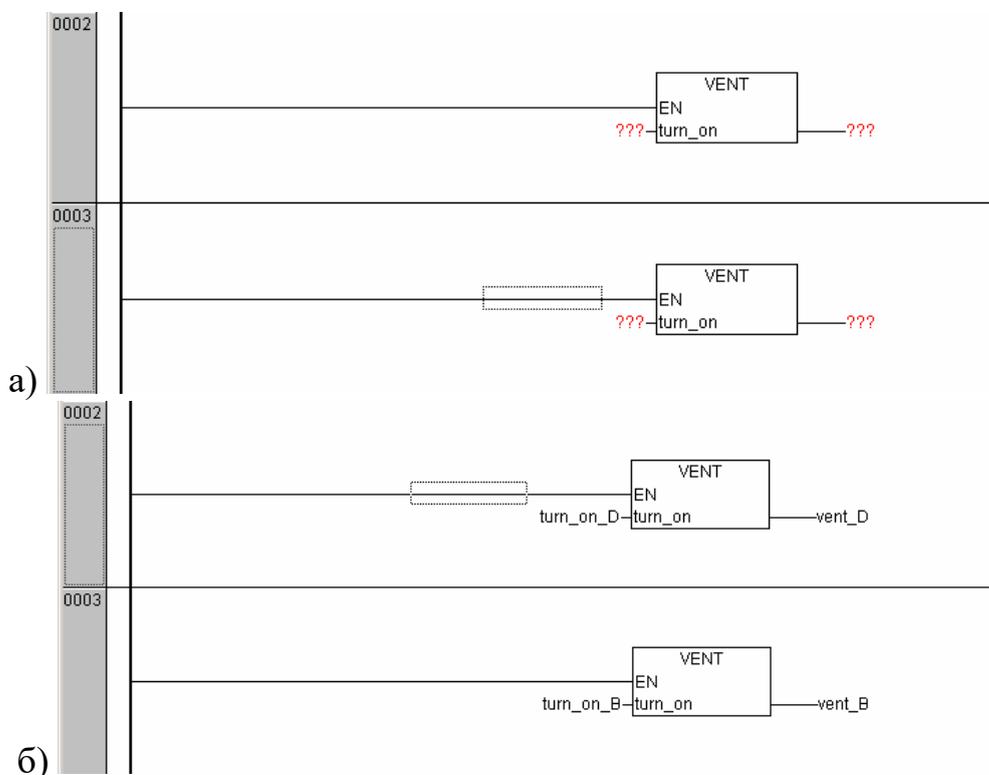


Рис. 26. Сегмент программы управления вентиляторами

7. Пример программирования контроллера ПЛК160. В качестве примера представлена программа управления движением тележки.

Тележка перемещается по прямолинейному пути. Имеется два фиксированных положения П0 и П1. Исходное положение тележки – П0.

При кратковременном нажатии на кнопку «Пуск» тележка движется вперед до положения П1, стоит в положении П1 5 с и возвращается назад в положение П0. В положении П0 немедленно возникает реверс привода тележки. Тележка вновь движется в положение П1, стоит в нем 5 с и возвращается в П0. В положении П0 вновь возникает команда на движение до положения П1 и т.д. после 10 циклов перемещения в положение П1 при возвращении тележки в положение П0 возникает сигнал «Конец цикла».

Для управления движением тележки следует предусмотреть счетчик (СЧ) со счетом до 10, задержку времени (таймер) для формирования задержанного сигнала П1[↑] и память P_П о нажатии на кнопку «Пуск».

$$P_{\text{П}} = (\text{Пуск} + P_{\text{П}}) \cdot KЦ$$

Команда на перемещение тележки вперед (В) возникает в положении П0, если отсутствует сигнал счетчика СЧ, и сохраняется при наличии сигнала памяти пуска

P_{II} до достижения положения PII . Этому описанию соответствует логическое уравнение

$$B = (PI0 \cdot \overline{CЧ} + B) \cdot \overline{PI1} \cdot P_{II}$$

Команда на перемещение тележки назад (H) возникает через 5 с нахождения в положении PII и сохраняется при наличии переменной P_{II} до достижения положения $PI0$. Тогда логическое уравнение для команды H имеет вид

$$H = (PI1^{\uparrow} + H) \cdot \overline{PI0} \cdot P_{II}$$

Сигнал конца цикла ($KЦ$) возникает в положении $PI0$ при наличии сигнала счетчика $CЧ$ о завершении 10 циклов перемещения в положение PII и сигнала P_{II} . Сигнал $KЦ$ сохраняется до очередного нажатия на кнопку «Пуск». Тогда логическое уравнение для рассматриваемого сигнала имеет вид

$$KЦ = (PI0 \cdot CЧ \cdot P_{II} + KЦ) \cdot \overline{Пуск}$$

В ниже приведенной программе (рис. 49) использован счетчик на вычитание. При нажатии на кнопку «Пуск» в счетчик вводится уставка 10. Счетчик считает число циклов перемещения в положение PII , при достижении этого положения число в счетчике уменьшается на единицу. Счетчик сбрасывается по сигналу конца цикла $KЦ$. Признаком окончания цикла является число в счетчике равное нулю, при котором на выходе счетчика «0». Поэтому в уравнениях команд B и $KЦ$ необходимо использовать инверсию сигнала $CЧ$ по отношению к ранее записанным уравнениям:

$$B = (PI0 \cdot \overline{CЧ} + B) \cdot \overline{PI1} \cdot P_{II}$$

$$KЦ = (PI0 \cdot CЧ \cdot P_{II} + KЦ) \cdot \overline{Пуск}$$

Сброс счетчика производится сигналом конца цикла $KЦ$.

В табл. 10 представлена адресация команд и сигналов для ОВЕН ПЛК160. Соответственно, на рис. 27 представлена программа управления движением тележки.

Таблица 10

Команда, сигнал	$PI0$	$PI1$	Пуск	B	H	$KЦ$	P_{II}	$CЧ$	$PI1^{\uparrow}$
Адрес на модуле	$I1$	$I2$	$I3$	$Q1$	$Q2$	$Q3$			
Адрес ПЛК	%IX0.0	%IX0.1	%IX1.0.0	%QX2.0	%QX2.1	%QX2.2	%MX0.0	%MX0.1	%MX0.2
Символьный адрес	$p0$	$p1$	<i>start</i>	<i>forw</i>	<i>rev</i>	<i>cycle</i>	<i>mem</i>	<i>count</i>	<i>tim</i>

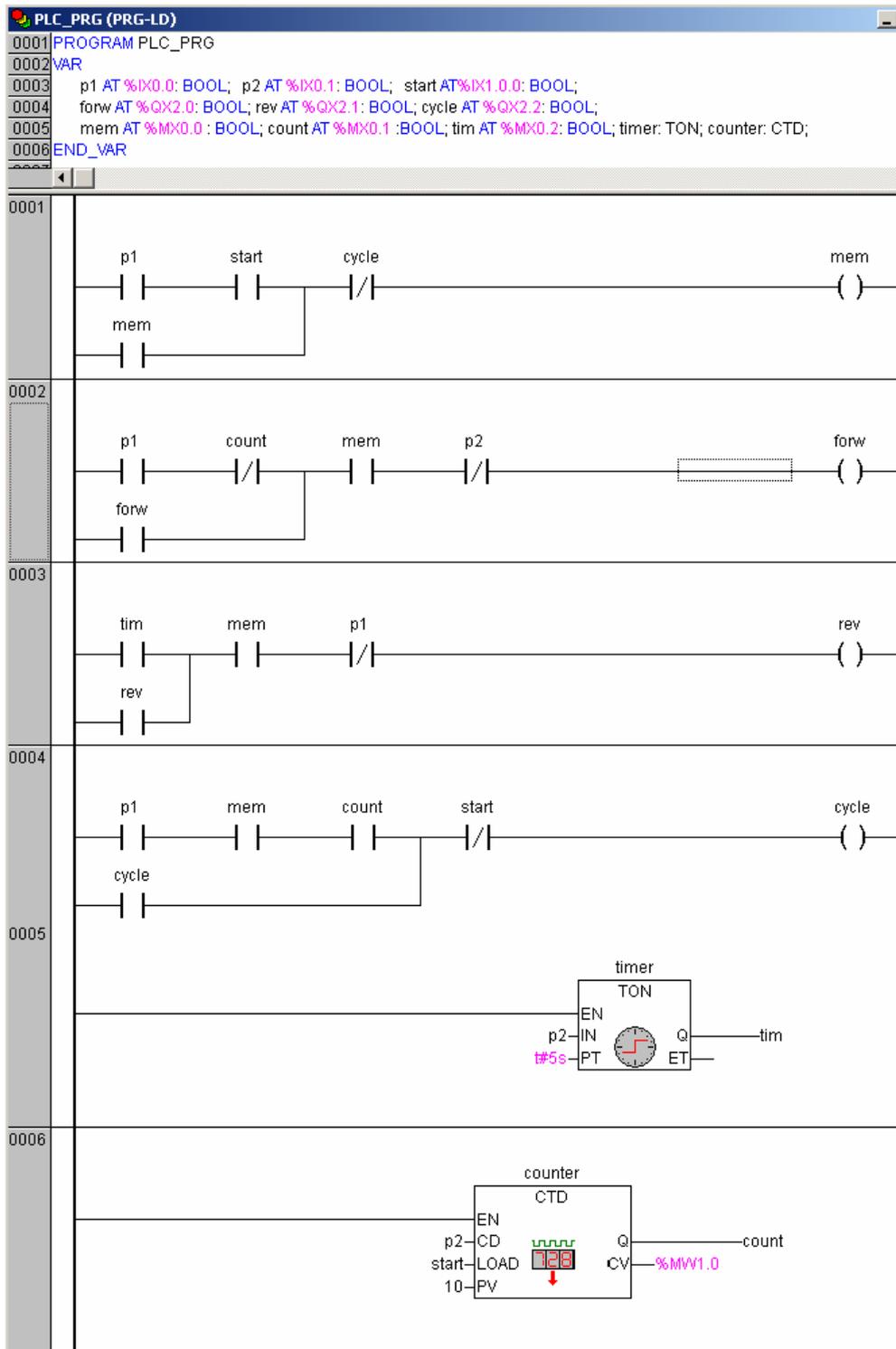


Рис. 27. Программа управления движением тележки

Загрузка программы в контроллер и запуск программы

Компиляция и загрузка программы в контроллер осуществляется в следующей последовательности:

- В выпадающем меню «Проект» выбрать пункт «Компилировать» либо воспользоваться «быстрой» клавишей *F11*. Пройдет компиляция проекта, по итогам

которой сформируется лог компиляции (см. рис. 16). Если ошибок в проекте нет, то весь текст будет серого цвета и в последней строке будет запись «0 ошибок, 0 предупреждений». В противном случае необходимо будет исправить все обозначенные ошибки.

– Настроить параметры порта связи с ПЛК160. Для этого в выпадающем меню «Онлайн» выбрать пункт «Параметры связи». В открывшемся окне нажать кнопку «New», в результате чего откроется окно создания нового канала связи (рис. 28а), в котором нужно выбрать строку *Serial (RS232)* и нажать «OK».

– В открывшемся окне настроек порта связи (рис. 28б) указать следующие значащие параметры: *Port – COM2, Baudrate – 9600, Parity – no, Stop bits – 1*.

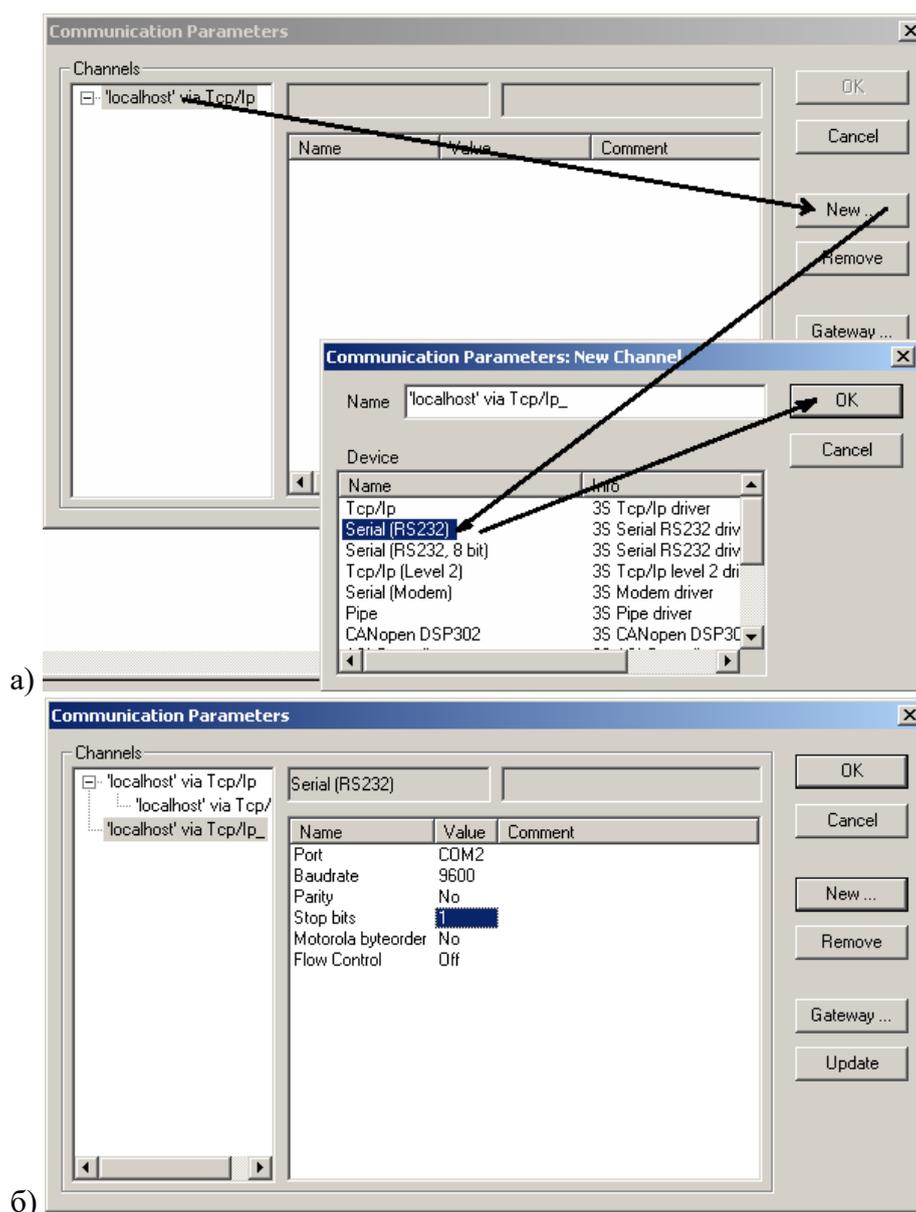


Рис. 28. Настройка параметров связи

– Создать загрузочный проект. Для этого в выпадающем меню «Онлайн» выбрать пункт «Создание загрузочного проекта». В итоге будут созданы файлы загрузочного проекта.

– Загрузить проект в память ПЛК. Для этого в выпадающем меню «Онлайн» выбрать пункт «Подключение», в появившемся окне (рис. 29а) выбрать «НЕТ». После этого в выпадающем меню «Онлайн» выбрать пункт «Загрузка». Прогресс процесса загрузки представлен на рис. 29б. В итоге проект полностью загружен в память ПЛК. Запуск контроллера осуществляется нажатием кнопки «СТАРТ/СТОП» либо выбором пункта «СТАРТ» в выпадающем меню «Онлайн»

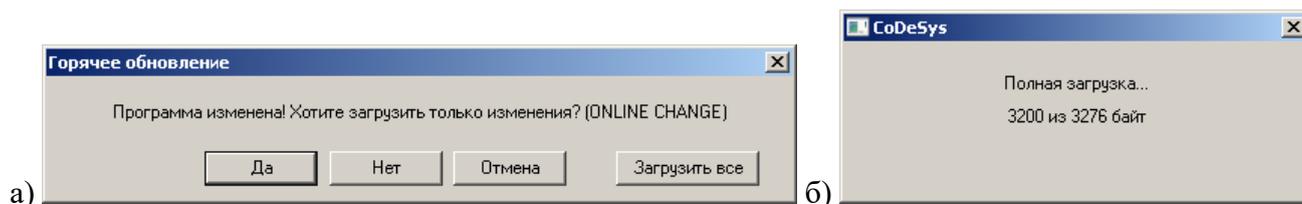


Рис. 29.

Во время работы контроллера доступен режим мониторинга хода выполнения программы. *CoDeSys* переходит в режим мониторинга сразу же после перехода в онлайн-режим работы. В этом режиме будут отображаться состояния контактов и обмоток, а также значения счетчиков, таймеров и других функции в реальном времени (рис. 30).

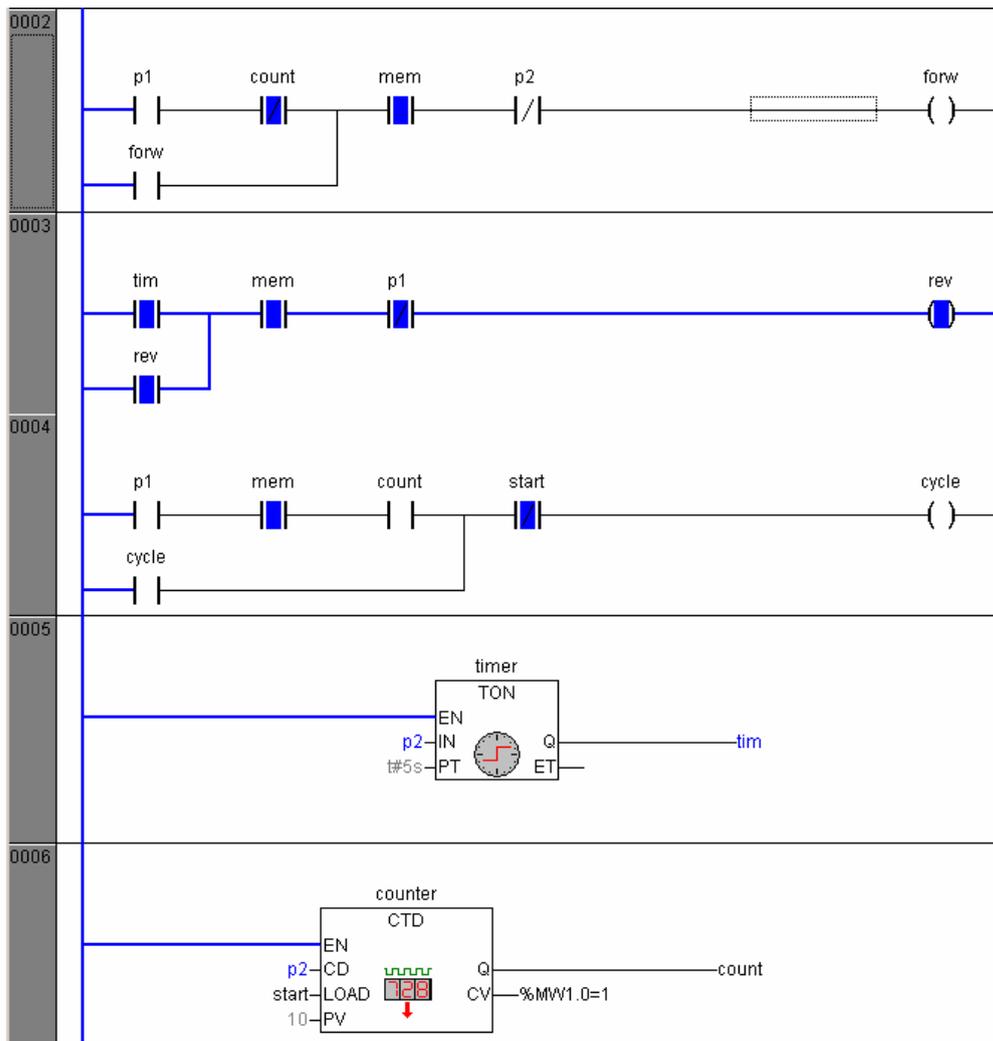


Рис. 30. Режим мониторинга

Варианты заданий

При подготовке к лабораторной работе студент должен в соответствии с табл. 11 выбрать свой вариант задания.

Для заданного варианта необходимо разработать алгоритм ее решения. В соответствии с заданием определиться с адресацией входных и выходных переменных. Затем определиться с адресацией вводимых промежуточных переменных и с адресацией используемых таймеров и счетчиков. Составить программу для контроллера ПЛК160.

Запустить программу в работу и убедиться в правильности ее работы. При наличии ошибок в работе устранить их и продемонстрировать преподавателю работу правильно функционирующей программы.

Таблица 11

Номер бригады	1 и 5			2 и 6			3 и 7			4 и 8		
Член бригады	1	2	3	1	2	3	1	2	3	1	2	3
Номер варианта	1	2	3	4	5	6	7	8	9	10	11	12

Вариант 1. Перекладыватель служит для перемещения листов со стола на постоянно вращающийся рольганг. Перекладыватель имеет подъемник листов, установленный на тележке. Подъемник представляет собой поперечину, перемещающуюся по вертикальным стойкам сверху вниз и обратно. Поперечина имеет на конце электромагнит для притягивания листов.

В исходном положении перекладыватель стоит в положении П0 над столом, а его подъемник в крайнем верхнем положении КВ. При поступлении листа на стол подается сигнал наличия листа, подъемник включается для движения вниз. При касании листа подъемник останавливается и включается электромагнит. Через 2 с включается подъемник на подъем, достигает положения КВ и останавливается. Включается тележка и перемещается в положение П1 над рольгангом, где останавливается и электромагнит отключается. Лист падает на рольганг. Через 2 с тележка движется в положение П0. Цикл повторяется при поступлении очередного сигнала наличия листа. Включение системы в работу – нажатием кнопки «Пуск».

Вариант 2. В контроллере программным путем необходимо реализовать генератор импульсов. Время наличия импульса 1 с, время его отсутствия 2 с. При нажатии на кнопку «Пуск» начинается счет импульсов. При прохождении 10 импульсов загорается лампа Л1, при прохождении следующих 10 импульсов – лампа Л2, аналогично включаются лампы Л3 и Л4. После загорания лампы Л4 (т.е. после прохождения 40 импульсов) счет прекращается, и все лампы горят до нажатия на кнопку «Стоп». Генератор импульсов реализовать программным путем.

Вариант 3. Автоматизировать перемещение пуансона штампованного пресса. Рабочий заправляет металлическую ленту (заготовку) и нажимает кнопку «Пуск». Пуансон из крайнего верхнего положения П0 движется вниз, при своем движении до положения п1 входит в матрицу, выбивая из ленты деталь необходимой формы, и возвращается в положение П0. чисто механическим устройством лента перемещается на один шаг и через 2 с пуансон совершает очередное движение.

Лента рассчитана на изготовление 10 деталей, поэтому через 10 ходов пуансона формируется сигнал «конец цикла». При подаче питания на систему автоматизации пуансон из любой точки должен прийти в положение П0.

Вариант 4. На программируемом контроллере реализовать систему управления толкателем с кривошипно-шатунным механизмом с реверсивным электродвигателем. Толкатель имеет два фиксированных положения П0 и П1. При подаче питания на систему толкатель из любого положения движется вперед медленно до положения П0. При появлении кратковременного сигнала «Пуск» толкатель из положения П0 медленно движется до положения П1, стоит там в течение 2 с, а затем электродвигатель включается для движения назад быстро и возвращается в положение П0, где формируется сигнал «Конец цикла». Цикл повторяется при нажатии кнопки «Пуск». При нажатии кнопки «Стоп» толкатель из любого положения на быстрой скорости возвращается в положение П0.

Вариант 5. Автоматизировать управление крышкой котла (типа большой кастрюли). При кратковременном нажатии кнопки «Открыть» крышка поднимается до крайнего верхнего положения, в котором поворачивается, открывая доступ к содержимому котла. При кратковременном нажатии на кнопку «Заккрыть» крышка поворачивается в положение точно над котлом и затем опускается, закрывая котел. Для исключения ударов предусмотреть переход на пониженную скорость перед касанием крышкой котла при его закрывании.

Вариант 6. Автоматизировать управление дверью (воротами) въезда/выезда гаража. Нормальное (исходное) положение двери закрытое. При кратковременном нажатии кнопки «Открыть» включается звонок и через 5 с включается привод двери на открывание. При полном открытии двери привод двери отключается сразу, а звонок звенит еще 2 с. Закрывание дверей – аналогично: при кратковременном нажатии кнопки «Заккрыть» включается звонок, через 5 с включается привод двери на закрывание. При полном закрытии двери привод двери отключается, а звонок звенит еще 2 с.

Вариант 7. Печь сопротивлений содержит три нагревательных элемента (НЭ), каждый из которых питается от собственного тиристорного преобразователя соответственно ТП1...ТП3. Сигнал «1» на входе ТП соответствует максимально допустимому току НЭ, сигнал «0» – минимально допустимому току НЭ. Синтезировать схему, обеспечивающую алгоритм работы группы ТП, представленный в табл. 12.

Таблица 12

Номер импульса генератора	-	1	2	3	4	5	6	7	8	9	...
Состояние ТП1	0	1	0	0	1	1	0	1	0	0	...
Состояние ТП2	0	0	1	0	0	1	1	0	1	0	
Состояние ТП3	0	0	0	1	1	0	1	0	0	1	

Начало работы осуществляется после нажатия на кнопку «Пуск». Цикл управления печью сопротивления состоит из 6 тактов. Переход с одного такта на другой осуществляется при поступлении очередного импульса от генератора импульсов. Циклы управления непрерывно следуют друг за другом. При

поступлении команды «Стоп» прекращается подача импульсов и все ТП отключаются. Для возобновления работы необходимо нажать на кнопку «Пуск». Генератор импульсов реализуется программным путем.

Вариант 8. Манипулятор (рука со схватом) служит для подачи заготовок из накопителя в пресс. Рука манипулятора может перемещаться вперед/назад из крайнего заднего КЗ в крайнее переднее КП положение и назад, а также поворачиваться из крайнего правого положения КПр в крайнее левое положение КЛ и обратно. Схват имеет электромагнитный механизм зажима/разжима заготовок. Исходное положение манипулятора перед накопителем (в положениях КЗ и КПр, схват под действием пружин зажат). При поступлении запроса со станка и наличии заготовки в накопителе включается электромагнит разжима схвата. Рука идет в крайнее переднее положение КП, через 2 с схват зажимает заготовку (электромагнит схвата отключается) и рука возвращается в положение КЗ. Рука поворачивается в положение КЛ и выдвигается вперед до КП. Схват разжимается (электромагнит схвата включается) и через 2 с рука идет в положение КЗ, затем поворачивается в положение КПр (исходное положение руки). Если есть запрос со станка, а накопитель пуст, то включается сигнальная лампа «Нет детали» и кратковременный звуковой сигнал (в течение 5 с). Световой сигнал снимается кнопкой «Сброс». Включение системы в работу – нажатием кнопки «Пуск».

Вариант 9. Автоматизировать процесс безупорной остановки длиномерного металла на рольганге Р1. На рольганг Р1 металл передается с предыдущего рольганга Р0, наличие металла на котором контролируется датчиком Д0. Вдоль рольганга Р1 расположены датчики, контролирующие наличие металла последовательно в положениях П1, П2 и П3. Нормальное положение рольганга Р1 – неподвижное. При наличии сигнала с датчика Д0 и отсутствии металла на рольганге Р1, рольганг Р1 включается, и металл перемещается по рольгангу Р1. При достижении передним концом металла положения П1 происходит снижение скорости Р1 до «ползучей» скорости, а в положении П2 привод рольганга Р1 отключается. Если по каким-либо причинам металл достиг положения П3, то привод рольганга Р1 реверсируется и работает на пониженной скорости, пока металл не уйдет из положения П3. При остановке переднего конца металла между положениями П2 и П3 формируется сигнал разрешения работы механизмов уборки металла с Р1. Включение системы в работу – при нажатии на кнопку «Пуск».

Вариант 10. Автоматизировать работу толкателя с кривошипно-шатунным механизмом с нереверсивным электродвигателем. При нажатии кнопки «Пуск» кривошип из положения $\alpha = 0^{\circ}$ движется до положения $\alpha = 120^{\circ}$. В этом положении происходит переход на пониженную (малую) скорость, кривошип идет до положения $\alpha = 180^{\circ}$ и останавливается. Через 3с электродвигатель автоматически вновь включается и на повышенной скорости шатун идет в положение $\alpha = 0^{\circ}$. Стоит в нем 3 с, включается и движется к положению $\alpha = 120^{\circ}$ и на пониженной скорости до $\alpha = 180^{\circ}$ и т.д. После 5 толканий толкатель останавливается в исходном положении (кривошип - в положении $\alpha = 0^{\circ}$) и формируется сигнал «Конец цикла».

Очередной цикл из пяти толканий – после очередного нажатия на кнопку «Пуск». При нажатии на кнопку «Стоп» толкатель из любого положения на повышенной скорости движется в положение $\alpha = 0^0$ и останавливается.

Вариант 11. Тележка движется по кольцевому пути, на котором есть 3 фиксированных положения П0, П1 и П2. Привод тележки – от нереверсивного электродвигателя. При подаче питания из любого положения тележка должна идти в положение П0. При нажатии на кнопку «Пуск» тележка без задержки движется в положение П1, стоит там в течение 3 с, затем движется в П2, стоит там в течение 4 с, движется в положение П0, стоит там в течение 5 с, движется в П1, стоит там в течение 3 с и т.д. То есть тележка непрерывно движется по кольцевому пути с остановками в фиксированных положениях. При нажатии на кнопку «Стоп» тележка должна без остановки идти в П0 и там ожидать очередного нажатия кнопки «Пуск».

Вариант 12. Автоматизировать линию сортировки изделий. По конвейеру движутся низкие изделия, но редко могут встречаться и высокие. По ходу конвейера установлены две фотоголовки на расстоянии друг от друга, равном половине ширины изделия. Первая по ходу конвейера фотоголовка настроена на низкие изделия (нижняя фотоголовка), вторая – на высокие изделия (верхняя фотоголовка). При прохождении низкой детали перекрывается только нижняя фотоголовка, а при высоких – вначале нижняя, а затем верхняя фотоголовки. На выходе конвейера детали ударяются о сортировочную планку, и высокие изделия поступают в правый накопитель (левое положение планки), низкие – в левый (правое положение планки). Планка перебрасывается пневмоприводом. Для переброски планки влево или вправо достаточно хотя бы кратковременно подать команду на ее перемещение влево или вправо. Включение системы в работу – нажатием кнопки «Пуск».

Требования к отчёту

Отчет должен содержать:

1. цель работы;
2. условия задачи по варианту, принятые обозначения переменных, логические функции для выходных и промежуточных переменных, при необходимости циклограмму работы оборудования;
3. логические функции в адресах программируемого контроллера;
4. программу для реализации системы управления;
5. методику экспериментальной проверки функционирования реализованной системы управления и результаты проверки;
6. выводы по работе.

Контрольные вопросы

1. Что означает термин «конфигурирование проекта»?
2. Что означает термин «*target*-файл», где и как он используется в программной среде *CoDeSys*?
3. Что такое объявление переменных и это используется при программировании контроллера?
4. Можно ли в ПЛК160 реализовать таймер с уставкой времени 0,07 с?
5. При каких условиях на выходе таймеров контроллера формируется сигнал «0» и сигнал «1»?
6. При каких условиях на выходе счетчиков контроллера формируется сигнал «0» и сигнал «1»?
7. Что означает создание функционального блока (*FB*) при программировании контроллера?
8. Что означает создание и открытие функции (*FC*) при программировании контроллера?

2.2 Работа №2. Основы работы со SCADA-системой

Цель работы

Ознакомиться с основами работы SCADA-системы *TRACE MODE* и приобрести навыки разработки исполнительской среды для осуществления обмена данными. Разработать систему автоматизации на базе SCADA-системы *TRACE MODE*.

Содержание работы

а) Изучить назначение и технические характеристики устройств автоматизации, входящих в состав лабораторного комплекса.

б) Изучить принципы конфигурирования функций обмена данными между SCADA-системой *TRACE MODE* и устройствами автоматизации, а также принципы программирования задач управления технологическими процессами.

в) Дома при подготовке к работе:

- составить последовательность, в которой будет вестись настройка и конфигурирование изучаемых устройств автоматизации;
- составить программу управления для ПЛК160;
- разработать проект для исполнительской среды SCADA-системы *TRACE MODE*.

г) В лаборатории:

- пройти тестирование по функциональным возможностям, режимам работы и основам конфигурирования оборудования, участвующего в обмене данными со SCADA-системой;
- пройти тестирование по системе команд и принципу программирования на ПЛК160;
- учитывая установленную при конфигурировании адресацию, откорректировать подготовленные дома программы, записать их в программной среде *CoDeSys 2.3* для ПЛК160/

Общие сведения

Краткие сведения о SCADA-системе *TRACE MODE*

SCADA (от англ. *Supervisory Control And Data Acquisition* – Диспетчерское управление и сбор данных) – программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления.

SCADA-система может являться частью АСУТП (автоматизированная система управления технологическим процессом), АСКУЭ (автоматизированная система коммерческого учета электроэнергии), системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д. *SCADA*-системы используются

во всех отраслях хозяйства, где требуется обеспечивать операторский контроль за технологическими процессами в реальном времени.

Данное программное обеспечение устанавливается на компьютеры и, для связи с объектом, использует драйверы ввода-вывода или *OPC/DDE* серверы. Программный код может быть как написан на языке программирования (например, на C++), так и сгенерирован в среде разработки.

Иногда *SCADA*-системы комплектуются дополнительным ПО для программирования промышленных контроллеров. Такие *SCADA*-системы называются интегрированными и к ним добавляют термин *SoftLogic*.

Термин *SCADA* имеет двойное толкование. Наиболее широко распространено понимание *SCADA* как приложения, то есть программного комплекса, обеспечивающего выполнение указанных функций, а также инструментальных средств для разработки этого программного обеспечения. Однако, часто под *SCADA*-системой подразумевают программно-аппаратный комплекс. Подобное понимание термина *SCADA* более характерно для раздела телеметрия.

Значение термина *SCADA* претерпело изменения вместе с развитием технологий автоматизации и управления технологическими процессами. В 80-е годы под *SCADA*-системами чаще понимали программно-аппаратные комплексы сбора данных реального времени. С 90-х годов термин *SCADA* больше используется для обозначения только программной части человеко-машинного интерфейса АСУ ТП.

Основные задачи, решаемые *SCADA*-системами:

- обмен данными с УСО (устройства связи с объектом, то есть с промышленными контроллерами и платами ввода/вывода) в реальном времени через драйверы;
- обработка информации в реальном времени;
- логическое управление;
- отображение информации на экране монитора в удобной и понятной для человека форме;
- ведение базы данных реального времени с технологической информацией;
- аварийная сигнализация и управление тревожными сообщениями;
- подготовка и генерирование отчетов о ходе технологического процесса;
- осуществление сетевого взаимодействия между *SCADA* и ПК;
- обеспечение связи с внешними приложениями (СУБД, электронные таблицы, текстовые процессоры и т. д.). В системе управления предприятием такими приложениями чаще всего являются приложения, относимые к уровню *MES*;
- *SCADA*-системы позволяют разрабатывать АСУ ТП в клиент-серверной или в распределенной архитектуре.

SCADA-система обычно содержит следующие подсистемы:

1) Драйверы или серверы ввода-вывода - программы, обеспечивающие связь *SCADA* с промышленными контроллерами, счётчиками, АЦП и другими устройствами ввода-вывода информации.

2) Система реального времени - программа, обеспечивающая обработку данных в пределах заданного временного цикла с учетом приоритетов.

3) Человеко-машинный интерфейс (*HMI*) - инструмент, который представляет данные о ходе процесса человеку оператору, что позволяет оператору контролировать процесс и управлять им. Программа-редактор для разработки человеко-машинного интерфейса.

4) Система логического управления - программа, обеспечивающая исполнение пользовательских программ (скриптов) логического управления в *SCADA*-системе. Набор редакторов для их разработки.

5) База данных реального времени - программа, обеспечивающая сохранение истории процесса в режиме реального времени.

6) Система управления тревогами - программа, обеспечивающая автоматический контроль технологических событий, отнесение их к категории нормальных, предупреждающих или аварийных, а также обработку событий оператором или компьютером.

7) Генератор отчетов - программа, обеспечивающая создание пользовательских отчетов о технологических событиях. Набор редакторов для их разработки.

8) Внешние интерфейсы - стандартные интерфейсы обмена данными между *SCADA* и другими приложениями. Обычно *OPC*, *DDE*, *ODBC*, *DLL* и т. д.

SCADA-система *TRACE MODE 6* (*Adastra Research Group, LTD*) предназначена для автоматизации промышленных предприятий, энергетических объектов, интеллектуальных зданий, объектов транспорта, систем энергоучета и т.д. Масштаб систем автоматизации, создаваемых в *TRACE MODE*, может быть любым – от автономно работающих управляющих контроллеров и рабочих мест операторов (*APM*) до территориально распределенных систем управления, включающих десятки контроллеров и *APM*, обменивающихся данными с использованием различных коммуникаций – локальной сети, интранета/интернета, последовательных шин на основе *RS232/485*, выделенных и коммутируемых телефонных линий, радиоканалов и *GSM/GPRS*-сетей. Причем, благодаря наличию в составе *TRACE MODE 6* компонентов *T-Factory.exe*, появляется возможность комплексной автоматизации управления как технологическими, так и бизнес-процессами производства для достижения высокой экономической эффективности и быстрого возврата инвестиций.

TRACE MODE 6 располагает встроенными драйверами, позволяющими подключать более 2500 наименований устройств ввода/вывода – программируемых логических контроллеров, удаленных *УСО*, плат ввода/вывода и промышленных сетей. Поддержка спецификаций *OPC DA* и *HDA*, протоколов *DDE* и *NetDDE*, а также открытый формат драйвера ввода/вывода и возможность прямого обращения к динамическим библиотекам (*DLL*) средствами языка программирования *ST* определяют широкие возможности по включению в состав систем автоматизации, разрабатываемых в *TRACE MODE*, разнообразного оборудования и обмену данными с внешними приложениями.

Системы, создаваемые в *TRACE MODE 6*, могут быть как информационно-измерительными (мониторинг), так и управляющими (*НЦУ*). Архитектура таких систем в свою очередь может быть как централизованной, так и распределенной, в зависимости от заданных требований.

Особое место отводится системам, использующим свободно-программируемые контроллеры (*PC-based* и/или *PAC*-контроллеры), поскольку в этом случае в *TRACE MODE 6* применяется единый инструмент создания информационного и математического обеспечения, как для АРМ верхнего уровня, так и для контроллеров, реализующих нижний уровень в иерархии систем автоматизации. Использование технологии автопостроения и подход к разработке проекта распределенной системы автоматизации как единого проекта существенно повышают производительность труда разработчиков систем, значительно уменьшая долю рутинных ручных операций и снижая количество ошибок, неизбежных в больших проектах.

Надежный и высокопроизводительный обмен данными между контроллерами и АРМ в *TRACE MODE 6* обусловлен использованием логического сетевого протокола *I-Net* (поверх *TCP/IP*), или *M-Link*. Хранение и доступ к накапливаемой информации реализуются через мощную систему архивирования технологических параметров СУБД РВ *SIAD 6*.

Динамические характеристики и надежность создаваемого в *TRACE MODE* программного обеспечения позволяют применять разработанные системы автоматизации в таких отраслях промышленности, как нефтехимия, металлургия, энергетика, машиностроение, коммунальное хозяйство, пищевая промышленность, транспорт, а также при проведении научных исследований.

Программная среда *TRACE MODE 6* подразделяется на интегрированную среду разработки и исполнительные модули (исполнительную среду).

Инструментальная система *TRACE MODE* (среда разработки) устанавливается на рабочем месте инженера-разработчика АСУ и предназначена для создания системы автоматизации и отладки всех ее компонентов. Сохраняемое в файл с расширением **.prj* описание создаваемой системы автоматизации является проектом *TRACE MODE*.

Исполнительные модули *TRACE MODE* (среда исполнения) предназначены для запуска проекта в реальном времени, т.е. для эксплуатации на действующем объекте автоматизации. Основным исполнительным модулем *TRACE MODE* для АРМ является монитор реального времени (МРВ), реализующий такие основные функции, как непрерывный сбор данных, их математическую обработку и визуализацию. Для запуска проекта в контроллерах используются исполнительные модули МикроМРВ, которые различаются по типу контроллеров (разрядность процессора, операционная система, использование сетевого взаимодействия и др.).

Каждому компьютеру/контроллеру, запускаемому под управлением исполнительного модуля в проекте *TRACE MODE*, сопоставлен отдельный узел. Максимальное количество узлов в проекте – 255.

В рамках узла создаются каналы – основные информационные единицы для ввода и первичной обработки данных. Каналы в рамках узла могут объединяться в группы, группы могут содержать подгруппы, образуя иерархическую информационную структуру произвольной вложенности. Каналы разделяются на классы в зависимости от типа обрабатываемых данных, например, для целочисленных – *HEX16* и *HEX32*, для вещественных – *FLOAT* и *DOUBLE FLOAT*.

Каналы содержат атрибуты. Атрибуты каналов могут быть вычисляемыми в реальном времени и не вычисляемыми, общими и специализированными, т.е. отражающими специфику класса канала.

Атрибуты канала, задаваемые при редактировании в ИС это Базовое имя, Комментарий, Кодировка. Они являются общими атрибутами каналов всех классов.

Атрибуты «Верхний предел», «Нижний предел» – не вычисляемые специализированные аргументы канала класса *FLOAT*, атрибут «Достоверность» является вычисляемым атрибутом, индицирующим в реальном времени состояние обмена данными с аппаратурой для каналов класса *HEX16* и *FLOAT*. Значения не вычисляемых атрибутов можно изменять в реальном времени, например, варьировать параметры внутренней обработки в канале класса *FLOAT* с помощью атрибутов «Множитель» и «Смещение» для выполнения линейного преобразования – перехода от кодов АЦП к физическим величинам и т.д.

Различают два типа каналов – *INPUT* и *OUTPUT*. В общем случае каналы типа *INPUT* могут получать информацию от источников данных, каналы типа *OUTPUT* – посылать управляющие воздействия в приемники данных.

Источники и приемники данных представляют собой описатели точек ввода-вывода, то есть связей с контроллерами, платами УСО, интеллектуальными датчиками и т.д. Каждая точка ввода-вывода может быть одним аналоговым сигналом или группой (до 16-ти) дискретных сигналов.

Связи с тэгами во внешних *OPC*-серверах и программными компонентами по протоколам *DDE/NetDDE* также относятся к точкам ввода-вывода.

Такие компоненты проекта как экраны, программы, связи с внешними реляционными СУБД и документы разрабатываются как шаблоны. Для связи шаблонов с атрибутами каналов используются аргументы. Вызов шаблонов в узлах проекта осуществляется с помощью специализированных каналов класса *CALL* (Вызов). Один шаблон может быть вызван многократно на разных узлах с передачей в аргументы различных атрибутов различных каналов. Допустимы связи между аргументами вызываемых шаблонов, если они заданы в рамках одного узла.

Для повторного использования в последующих проектах любых компонентов проекта – шаблонов экранов, программ, связей с СУБД, документов, источников/приемников и узлов в целом предназначена пользовательская библиотека.

Количество проектов, разрабатываемых с помощью одной инструментальной системы *TRACE MODE*, как и время работы в ней не ограничены.

В состав инструментальной системы *TRACE MODE 6* входят:

- 1) Интегрированная среда разработки *TRACE MODE 6 IDE* (файл *tmdevenv.exe*).
- 2) Профайлеры – отладочные МРВ (файл *rtc.exe* – с поддержкой графических экранов. Файл *rtmg32.exe* – без поддержки графических экранов).
- 3) Бесплатный набор драйверов устройств ввода\вывода.
- 4) Библиотека компонентов – файл *tmdevenv.tmul* и набор ресурсов – обои, логотипы, анимации в каталоге *\Lib* папки инструментальной системы.
- 5) Электронная документация (встроенная справочная система).

б) Демонстрационные проекты *TRACE MODE* (примеры систем автоматизации, основанные на имитации технологического процесса).

Создание проекта АСУ осуществляется в единой интегрированной среде разработки (ИС) *TRACE MODE 6*, работающей под управлением операционной системы *MS Windows*. Операции по созданию компонентов проекта, их редактированию и установлению взаимосвязей между ними выполняются в навигаторе проекта ИС. Разрабатываемый проект представляется в виде дерева компонентов. Создание дерева проекта облегчается применением различных технологий автопостроения.

При создании программ, экранов, связей с СУБД, документов и других компонентов вызываются соответствующие редакторы. Разработанные шаблоны могут быть применены для повторного использования, как в текущем проекте, так и в последующих при условии их сохранения в пользовательской библиотеке компонентов.

Конечным результатом работы инструментальной системы *TRACE MODE 6* является набор файлов, предназначенных для исполнения задач АСУ в мониторах реального времени на АРМ и в контроллерах. В рассматриваемых далее случаях в качестве МРВ для АРМ будет использоваться профайлер с поддержкой графических экранов *rtc.exe*, а для контроллера – без поддержки графических экранов *rtmg32.exe*, расположенные в директории инструментальной системы *TRACE MODE 6*. Профайлер позволяет запускать на компьютере с установленной инструментальной системой один узел разработанного проекта. Узлы проекта, запущенные в отладчиках *TRACE MODE 6* на разных компьютерах, могут обмениваться данными как по сети, так и по последовательным коммуникациям. На одном компьютере с несколькими сетевыми адаптерами, соединенными через *hub* или *switch*, можно одновременно запускать соответствующее количество узлов, обменивающихся данными по сети. На одном компьютере с несколькими последовательными портами, соединенными соответствующим образом (для двух – нуль-модемным кабелем), можно запускать также несколько узлов, обменивающихся данными по интерфейсам *RS232/485*.

Порядок выполнения лабораторной работы

В лабораторной работе студентам предлагается вариант системы автоматизации *SCADA*-система *TRACE MODE* + программируемый контроллер ПЛК160.

Порядок выполнения лабораторной работы заключается в следующем:

- 1) Конфигурирование устройства и разработка управляющей программы.
- 2) Разработка и настройка в инструментальной среде следующих компонентов:
 - системы визуализации (экрана контроля и управления);
 - области «Источники/приёмники»;
 - области «Каналы»;
 - привязка источников к каналам.
- 3) Создание исполнительного файла для среды исполнения.
- 4) Отладка.

Порядок разработки системы автоматизации на базе SCADA-системы TRACE MODE и программируемого контроллера ПЛК160:

1) Программирование ПЛК160 в программной среде CoDeSys 2.3. Выполняется в соответствии с методическими указаниями к лабораторной работе №5.

Так как в рассматриваемом варианте обмен информацией между SCADA-системой TRACE MODE и программируемым контроллером ПЛК160 осуществляется посредством использования OPC-сервера, то в программном блоке POU необходимо только определить переменные, которые будут участвовать в обмене данными (рис. 31).

2) Далее необходимо настроить конфигурацию проекта для совместной работы с OPC-сервером CoDeSys 2.3:

– на вкладке «Ресурсы» найдите пункт «Настройки целевой платформы» (рис. 32);

```
0001 PROGRAM PLC_PRG
0002 VAR
0003   Din AT %IX0.0:BOOL;
0004   Dout AT %QX2.0:BOOL;
0005   Ain AT %IB1.0:BYTE;
0006   Aout AT %QB3.0:BYTE;
0007 END_VAR
```

Рис. 31. Определение переменных программы

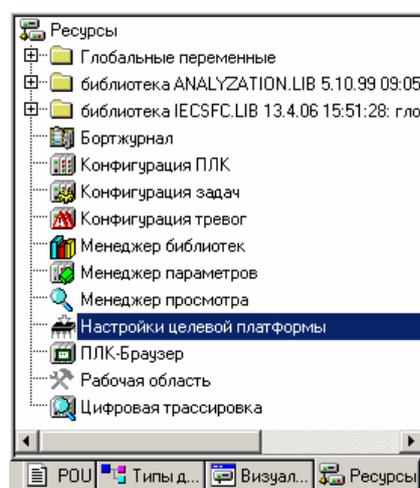


Рис. 32

– в появившемся окне в вкладке «Общие» установить галочку напротив пункта «Загружать символьный файл» (рис. 33);

– выбрать в меню «Проект» пункт «Опции». В разделе «Символьная конфигурация» установить галочку напротив пункта «Создавать описания»;

– нажать кнопку «Настроить символьный файл». В открывшемся окне «Установка атрибутов объекта» выделить те переменные, с которыми нужно будет работать в SCADA-системе, и нажать кнопку «ОК» (рис. 34);

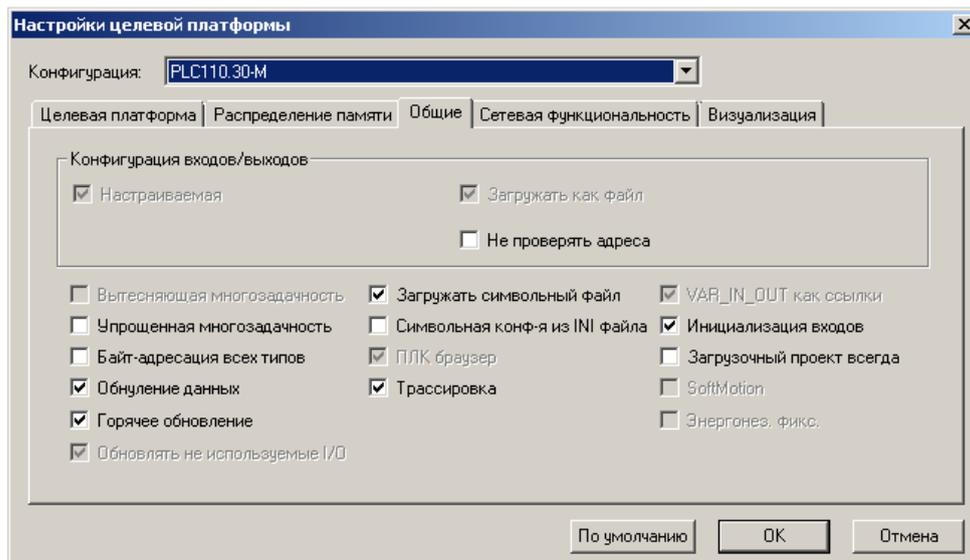


Рис. 33

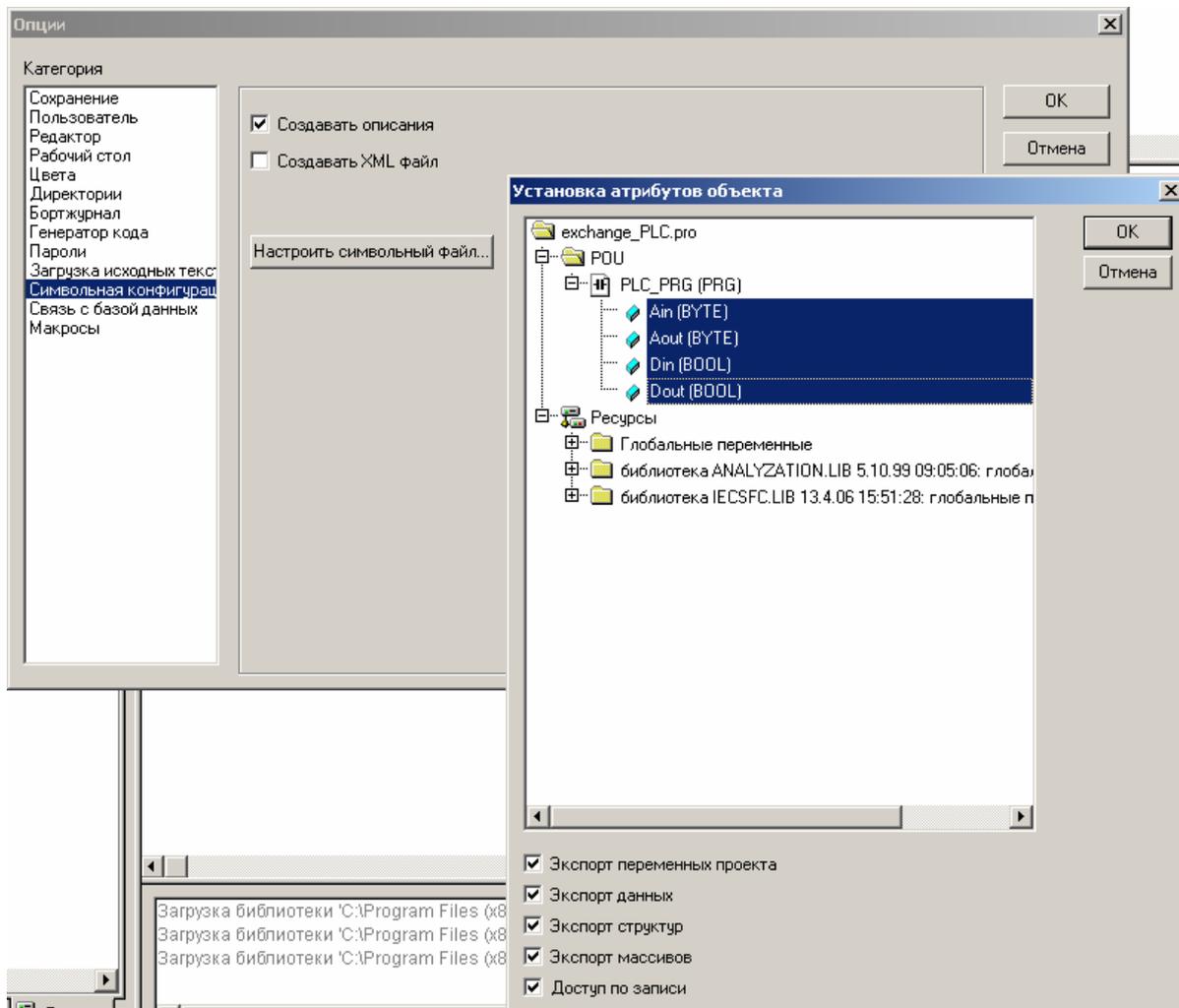


Рис. 34

– выполнить компиляцию проекта. Для этого выбрать в меню «Проекты» пункт «Компилировать все»;

– сохранить проект. Затем связаться с ПЛК, выбрав пункт «Подключение» в меню «Онлайн». После установки связи в меню «Онлайн» выбрать пункт «Создать загрузочный проект».

3) Конфигурирование OPC-сервера с помощью программы *OPC Configurator*:

– запустить программу *CoDeSys OPC Configurator* (рис. 35а). Это можно сделать через кнопку «Пуск» в Windows (Пуск → Программы → 3S Software → Communication → *CoDeSys OPC Configurator*);

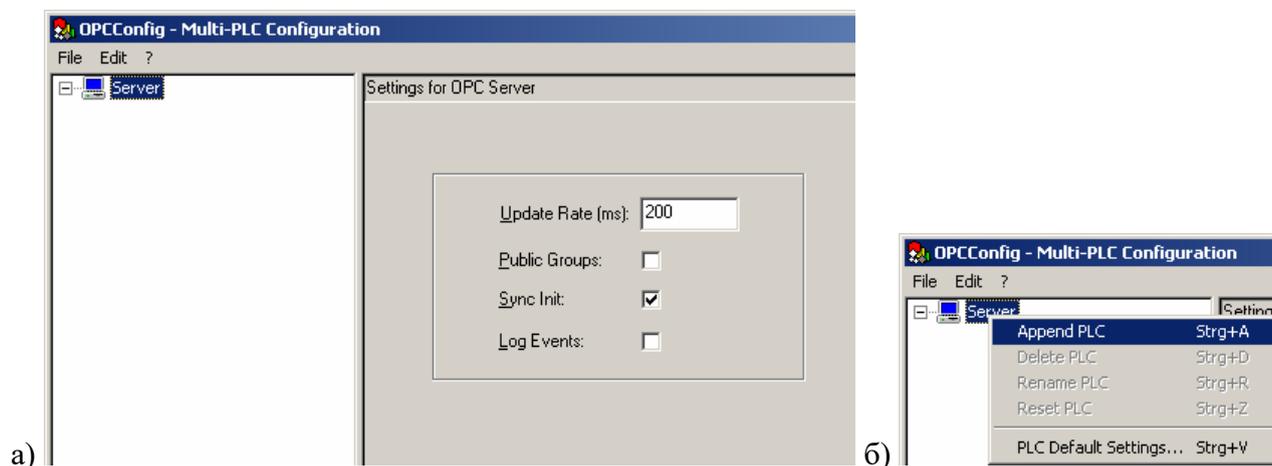


Рис. 35

– нажать правой кнопкой мыши на надписи «Server». В появившемся контекстном меню необходимо выбрать пункт «Append PLC» (рис. 35б). После этого в окне добавится значки  PLC1 и  Connection;

– выделить  PLC1 и в появившемся поле ввода «Project name» ввести имя проекта, который был загружен в ПЛК (рис. 36);

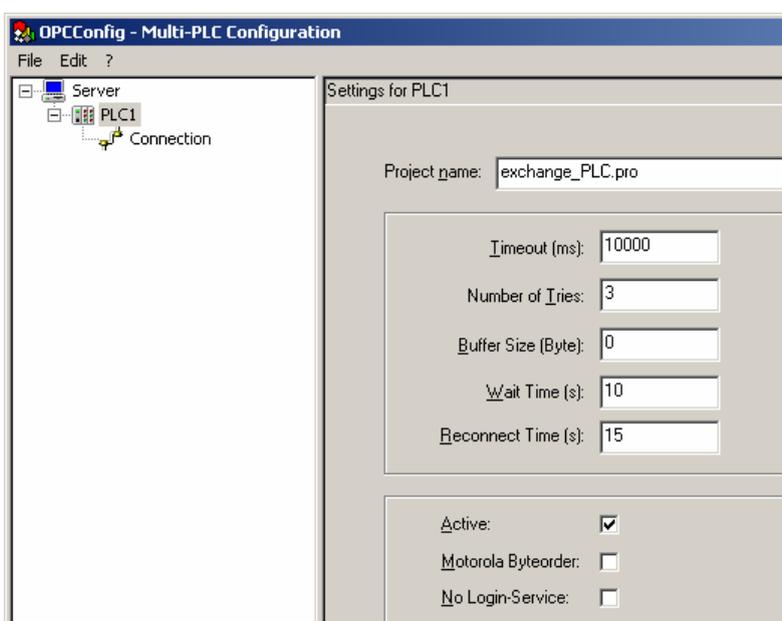


Рис. 36

– выделить значок  Connection и нажать кнопку «Edit» (рис. 37). В появившемся окне необходимо выбрать интерфейс и параметры сетевого обмена между ПЛК и компьютером. Эта процедура аналогична настройке сетевого обмена с ПЛК, описанной в методических рекомендациях к лабораторной работе по изучению программируемого контроллера ПЛК160. После выполнения всех необходимых действий таблица, приведенная на рис. 37, будет заполнена значениями параметров сетевого обмена. После этого программу *CoDeSys OPC Configurator* можно закрыть. После выполнения всех этих действий данные из OPC-сервера можно использовать в SCADA-системе.

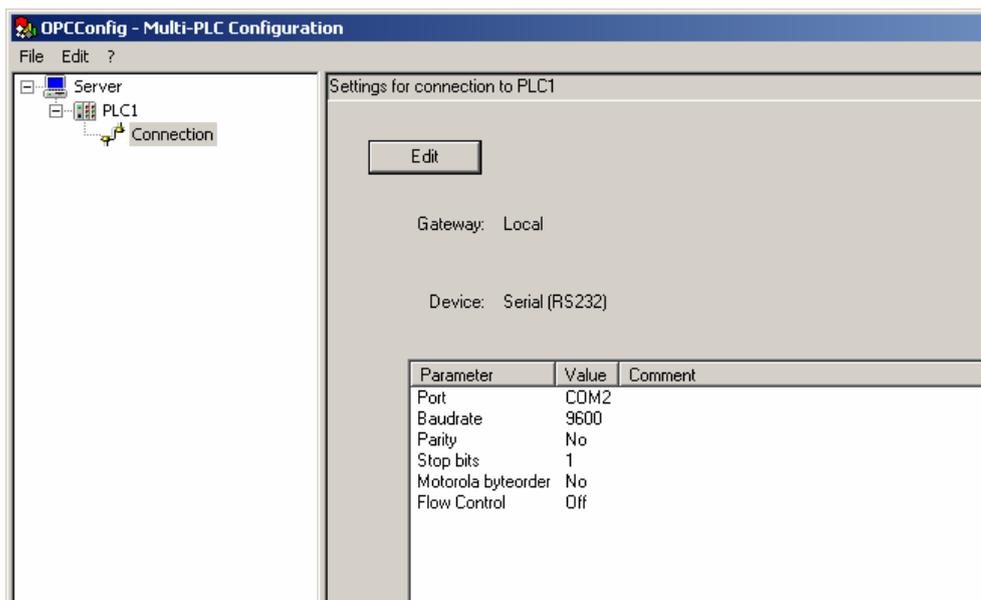


Рис. 37

4) Разработка исполнительной среды (исполнительного файла) SCADA-системы *TRACE MODE* выполняется в следующей последовательности:

– запустить инструментальную систему *TRACE MODE* б двойным щелчком левой кнопки мыши по иконке «*TRACE MODE*» рабочего стола *Windows* или из меню «Пуск»;

– в открывшемся окне выбрать *Файл* → *Новый*;

– в окне навигатора проекта представлены слои проекта. В слое «Система» создать новый узел *RTM_1*, нажав правую клавишу мыши и выбрав в открывшемся меню *Новый узел* → *RTM*;

– в созданном узле установить один экран. Для этого необходимо нажать правую клавишу мыши на узле *RTM_1* и выбрать *Создать компонент* → *Экран*. Окно навигатора проекта примет вид, представленный на рис. 38;

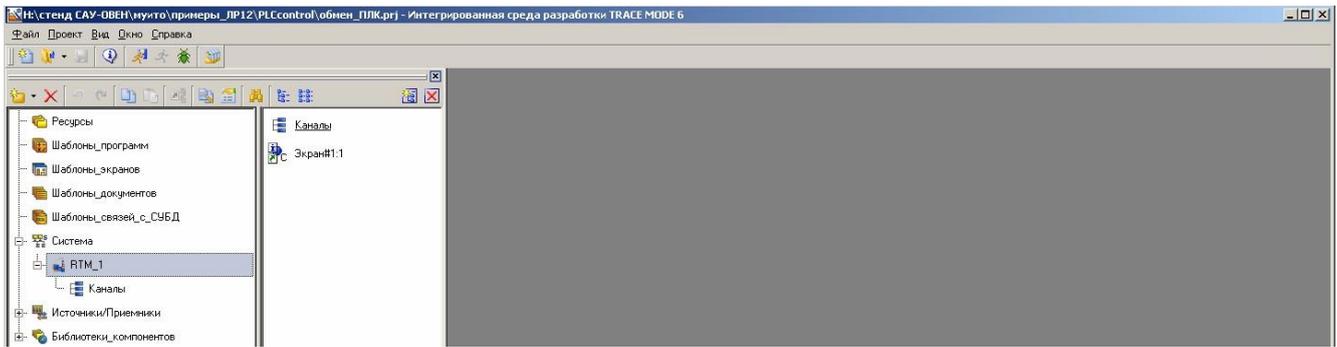


Рис. 38. Внешний вид проекта

– дважды щелкнув левой клавишей мыши по элементу «Экран#1:1» открыть созданный экран. В нем, используя графические элементы «Кнопка» , выключатель , плоские фигуры  и «Текст» , создать экран для визуализации обмена между SCADA-системой TRACE MODE и программируемым контроллером ПЛК160 (рис. 39);

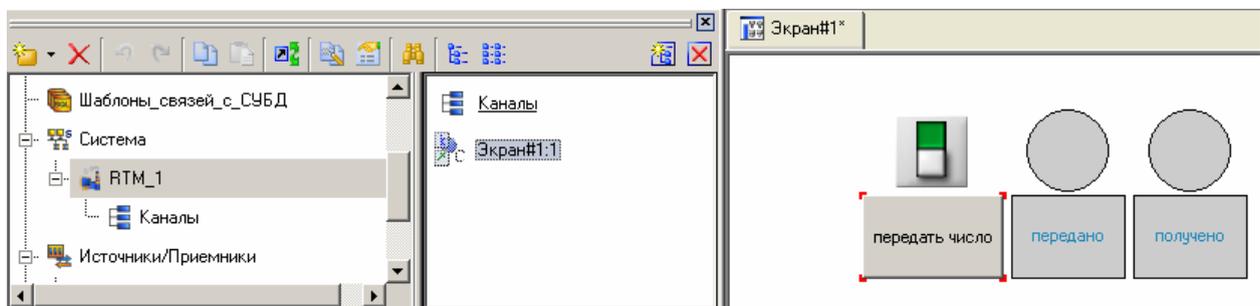
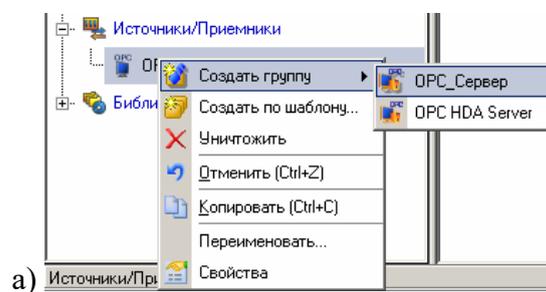


Рис. 39. Экран визуализации обмена между SCADA-системой и контроллером

– в слое «Источники/приемники» создать группу «OPC_1», для чего необходимо выбрать «Источники/приемники» и, нажав правой клавишей мыши, в открывшемся окне выбрать *Создать группу* → *OPC*;

– в группе «OPC_1» создать группу «OPC Сервер» (рис. 40а). Для этого необходимо выбрать «OPC_1» и, нажав правой клавишей мыши, в открывшемся окне выбрать *Создать группу* → *OPC Сервер*;



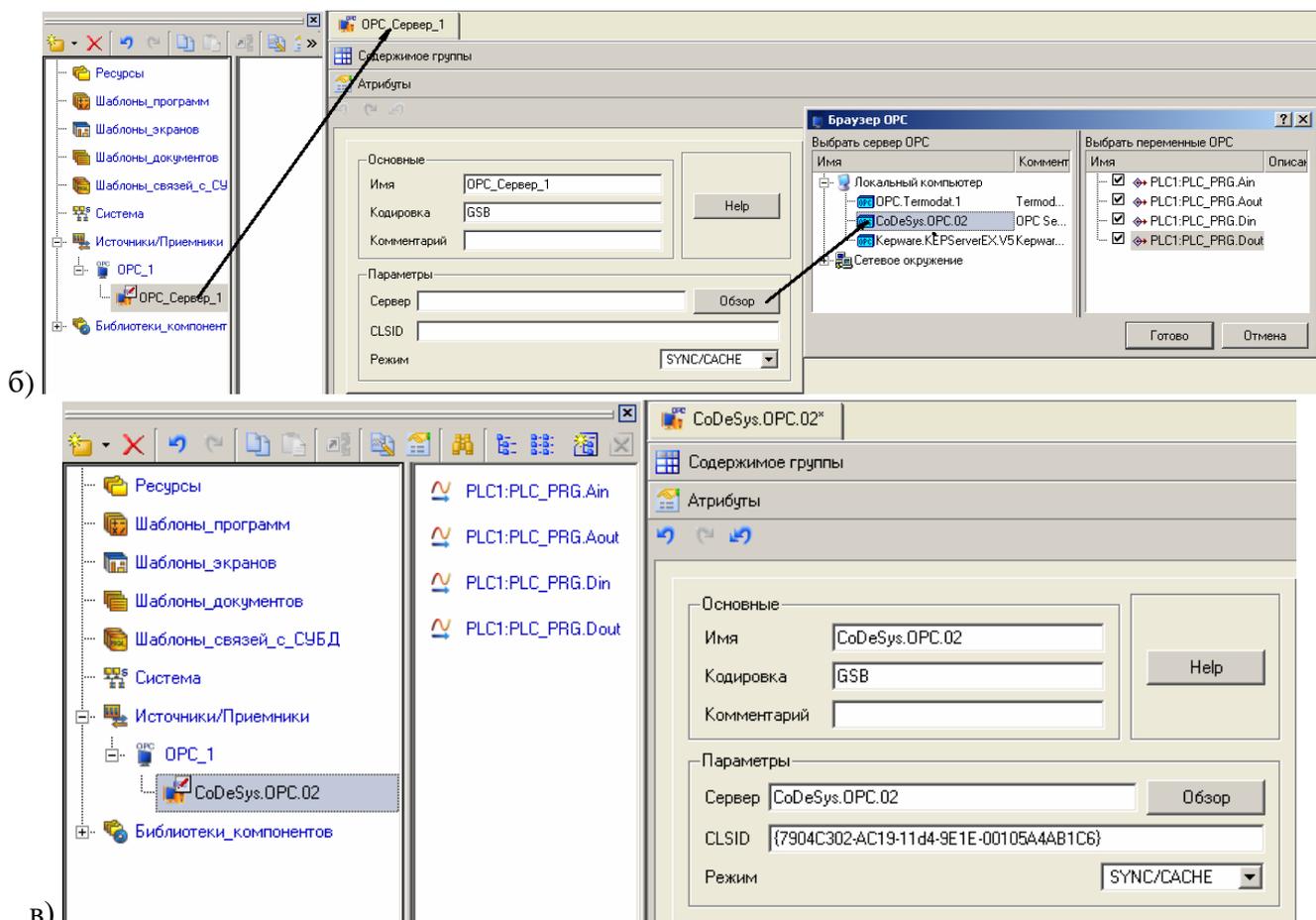


Рис. 40. Создание точек ввода/вывода в слое «Источники/приемники»

– правой клавишей мыши нажать на группе «*OPC_Сервер_1*» и в выпадающем меню выбрать пункт «Редактировать». В окне «Содержимое группы» (рис. 40б) в строке «Сервер» нажать кнопку «Обзор» и в открывшемся окне выбрать требуемый сервер *OPC* – *CoDeSys.OPC.02*. При выборе сервера в правой части окна отобразятся все созданные ранее переменные *OPC*. Необходимо установить галочку напротив требуемой переменной, после чего нажать «Готово». В результате этих действий в группе «*CoDeSys.OPC.02*» будет создано 4 точки ввода/вывода (рис. 40в);

– настроить свойства созданных точек ввода/вывода, для чего два раза щелкнуть левой клавишей мыши по необходимой точке ввода/вывода и в открывшемся окне задать настройки для точки в соответствии с табл. 13;

Таблица 13

№	Точка	Направление	Формат
1	<i>PLC1:PLC_PRG.Ain</i>	<i>Input</i>	Аналог
2	<i>PLC1:PLC_PRG.Aout</i>	<i>Output</i>	Аналог
3	<i>PLC1:PLC_PRG.Din</i>	<i>Input</i>	Дискрет
4	<i>PLC1:PLC_PRG.Dout</i>	<i>Output</i>	Дискрет

– создать каналы. Для этого для удобства необходимо открыть дополнительное окно навигатора, нажав кнопку «Открыть дополнительное окно навигатора» .

Используя указатель мыши последовательно перетащить созданные точки ввода/вывода в группу «Каналы» узла *RTM_1* (рис. 41);

– открыв экран, в меню «Вид» выбрать «Аргументы экрана». В открывшееся окно последовательно перетащить созданные каналы (см. рис. 41), после чего созданные аргументы экрана привязать к элементам экрана, последовательно перетащив аргументы на изображения элементов в соответствии с табл. 14;

Таблица 14

№	Канал	Аргумент	Элементы экрана
1	<i>PLC1:PLC_PRG.Ain</i>	<i>PLC1:PLC_PRG.Ain_R</i>	Текст «получено»
2	<i>PLC1:PLC_PRG.Aout</i>	<i>PLC1:PLC_PRG.Aout_R</i>	Кнопка «передать число», текст «передано»
3	<i>PLC1:PLC_PRG.Din</i>	<i>PLC1:PLC_PRG.Din_R</i>	Объект «л2»
4	<i>PLC1:PLC_PRG.Dout</i>	<i>PLC1:PLC_PRG.Dout_R</i>	Выключатель, объект «л1»

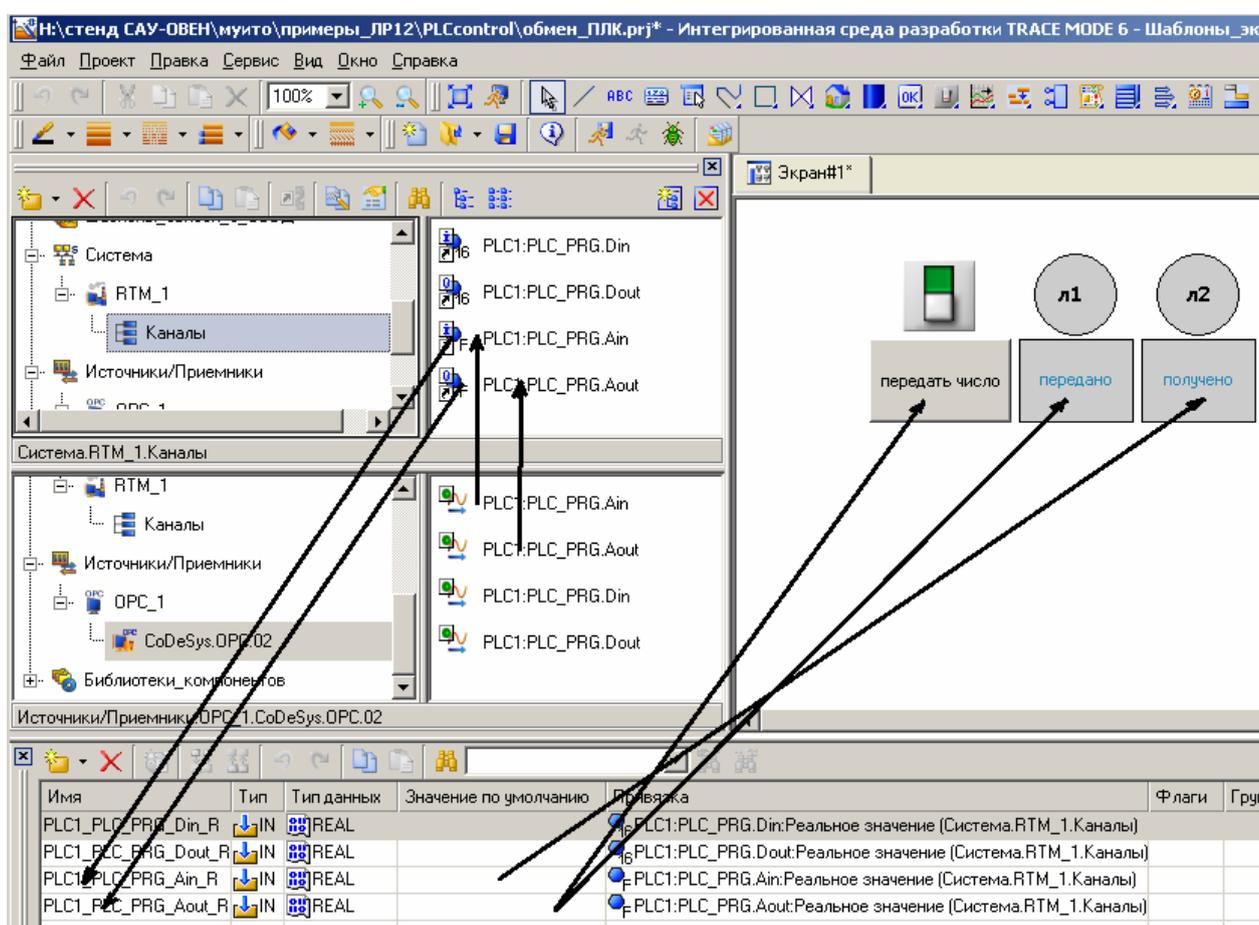


Рис. 41. Создание каналов и их привязка

– настроить свойства элементов экрана. Для этого дважды щелкнуть левой клавишей мыши по требуемому элементу. В открывшемся окне свойств для кнопки выбрать закладку «События» (рис. 42а), для выключателя, эллипса и текстового поля – «Общие свойства» (рис. 42б – 42г). Для кнопки необходимо настроить свойства передачи в закладке «Передать значение» в соответствии с табл. 14, для ламп – таблица 15, а для текстового поля в закладке «Текст» проверить правильность привязки элемента экрана и канала в соответствии с табл. 16;

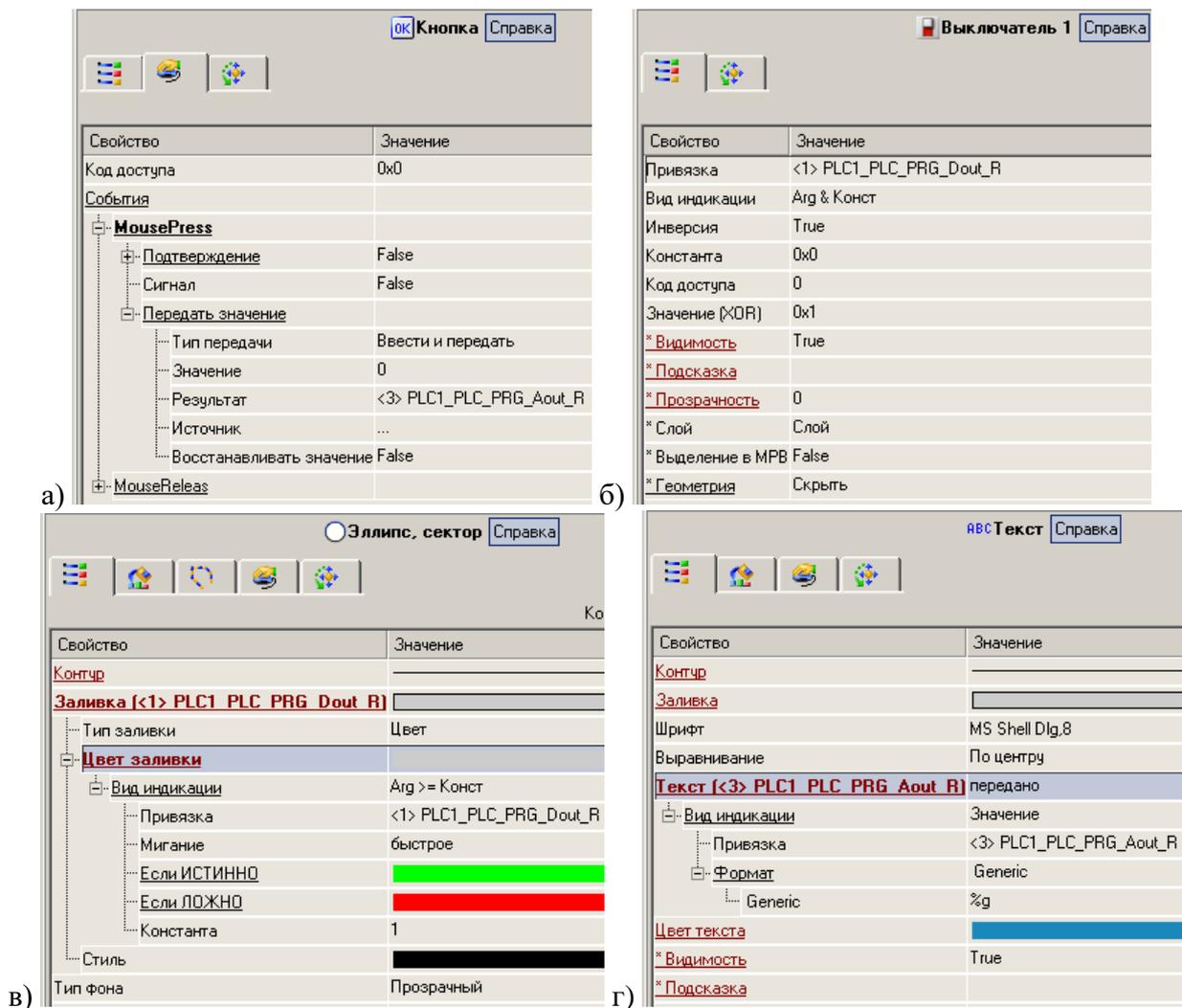


Рис. 42. Настройка свойств графических элементов

Таблица 14

№	Графический элемент	Тип передачи	Результат	Значение
1	Кнопка «передать число»	Ввести и передать	<i>PLC1:PLC_PRG.Aout_R</i>	0

Таблица 15

№	Графический элемент	Функция	Вид индикации	Привязка	Константа
1	Объект «л1»	Заливка	<i>Arg = Const</i>	<i>PLC1:PLC_PRG.Dout_R</i>	1
2	Объект «л2»	Заливка	<i>Arg = Const</i>	<i>PLC1:PLC_PRG.Din_R</i>	1

Таблица 16

№	Графический элемент	Функция	Вид индикации	Привязка	Формат
1	Текст «передано»	Текст	Значение	<i>PLC1:PLC_PRG.Aout_R</i>	<i>generic, %g</i>
2	Текст «получено»	Текст	Значение	<i>PLC1:PLC_PRG.Din_R</i>	<i>generic, %g</i>

– выделить узел *RTM_1* и нажать кнопку «Сохранить для МРВ»  на панели меню для создания исполнительного файла для монитора реального времени (МРВ). В результате выполнения этой процедуры в папке с фалом проекта должна появиться папка с таким названием, в которой располагается папка «*RTM_1*». В этой папке находится исполнительный файл для профайлера (исполнительной среды), имеющий имя «*Имя файла_0*».

3) Проверка правильности работы системы выполняется в следующей последовательности:

– подать питание на лабораторный стенд и используемые в лабораторной работе модули. При необходимости перевести программируемый контроллер ПЛК160 в режим «РАБОТА», нажав соответствующую кнопку на лицевой панели контроллера;

– запустить исполнительный файл дважды щелкнув по названию файла. В открывшемся окне для запуска профайлера необходимо нажать кнопку «Запуск» .

– используя кнопки на экране монитора, передать дискретное или числовое значение. На лицевой панели модуля должно отобразиться изменение состояний выходов контроллера. Аналогично проверить получение сигналов от контроллера: включая тумблеры на лицевой панели модуля, наблюдать изменение отображаемых значений в соответствующих числовых полях;

– при необходимости исправить ошибки, заново сохранить, нажав кнопку «Сохранить для МРВ» и проверить в профайлере.

Требования к отчёту

Отчет должен содержать:

1. цель работы;
2. описание технологической задачи по варианту, конфигурацию или технологическую схему автоматизированного объекта, обозначение переменных;
3. параметры программируемого контроллера введенные в него при конфигурировании;
4. проект для инструментальной среды SCADA-системы с комментариями.
5. описание методики и экспериментальной проверки работоспособности системы автоматизации;
6. выводы по работе.

Контрольные вопросы

1. Что означает термин «конфигурирование контроллера»?
2. В какой последовательности осуществляется конфигурирование ПЛК160?
3. В какой последовательности осуществляется разработка SCADA-проекта?
4. Как осуществляется обмен информацией между программируемым контроллером и SCADA-системой?
5. Как осуществляется конфигурирование OPC-сервера?
6. Как в лабораторной работе проверить правильность работы разработанной системы автоматизации?

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



УТВЕРЖДАЮ
И. о. проректора по учебно-методической работе
В. В. Зубов

**Методические указания
к выполнению лабораторных работ
Часть 2**

**Б1.В.09.08 МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В МЕХАТРОНИКЕ И
РОБОТОТЕХНИКЕ**

Специальность
21.05.04 Горное дело

Специализация
Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой

(подпись)

Волков Е. Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

(Дата)

Рассмотрены методической комиссией
факультета

Горно-механического

(название факультета)

Председатель

(подпись)

Осипов П. А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

ОГЛАВЛЕНИЕ

1. ОБЩЕЕ ОПИСАНИЕ СТЕНДА.....	4
1.1 Назначение стенда.....	4
1.2 Состав	5
1.3 Технические характеристики стенда.....	5
2. ЛАБОРАТОРНЫЕ РАБОТЫ.....	6
2.1 Работа №1. Изучение бесконтактных датчиков технологической информации	6
2.2 Работа №2. Изучение дискретной системы управления двигателем постоянного тока на основе программируемого логического контроллера	10
2.3 Работа №3. Синтез и анализ алгоритмов управления мехатронной станцией сборки деталей.....	14

1. ОБЩЕЕ ОПИСАНИЕ СТЕНДА

1.1 Назначение стенда

Лабораторный стенд предназначен для изучения программируемого логического контроллера с объектом управления в виде мехатронного модуля – сборки деталей. Стенд позволяет изучить основы построения систем цикловой автоматики. Элементная база мехатронного модуля состоит из ленточного конвейера с электроприводом постоянного тока, модуля подачи базовых деталей, модулей подачи крышек из различного материала, пневматических толкателей и датчиков различного типа.

Внешний вид учебного комплекта представлен ниже (см. Рисунок 1).



Рисунок 1 – Внешний вид стенда ММ-СБ-НН

1.2 Состав

Лабораторный стенд включает в себя следующие компоненты (см. Таблица 1).

Таблица 1 – Состав

№	Наименование	Кол-во
1	Моноблок, содержащий: источник питания, программируемый логический контроллер, элементы индикации и управления	1 шт.
2	Мехатронный модуль сборки деталей	1 шт.
3	Модуль пульта симуляции	1 шт.
4	Компрессор	1 шт.
5	Ноутбук	1 шт.
7	Комплект кабелей и принадлежностей для проведения лабораторных работ	1 шт.
8	Техническое описание	1 шт.
9	Методические указания к проведению лабораторных работ	1 шт.

1.3 Технические характеристики стенда

Общие технические характеристики типового комплекта учебного оборудования представлены ниже (см. Таблица 2).

Таблица 2 – Технические характеристики

Параметр	Значение
Напряжение электропитания, В	220
Частота питающего напряжения, Гц	50
Потребляемая мощность, ВА	1600
Габаритные размеры, мм	480x600x650
Масса, кг	60 кг
Диапазон рабочих температур	+10...+35 °С
Относительная влажность воздуха, не более	80%

2. ЛАБОРАТОРНЫЕ РАБОТЫ

2.1 Работа №1. Изучение бесконтактных датчиков технологической информации

Цель работы

Приобретение опыта работы с устройствами детектирования, идентификации и управления.

Содержание работы

Дома, при подготовке к работе:

- Изучить назначение, технические характеристики пульта управления, основные узлы и возможности лабораторного стенда;
- Изучить описание мехатронного модуля, ознакомиться с характеристиками всех его элементов;
- Уделить особое внимание датчикам, которые применяются в мехатронном модуле;

В лаборатории:

- Закрепить полученные при подготовке к лабораторной работе знания на практике;

Общие сведения

Мехатронный модуль представляет собой набор узлов, пневматических и электрических компонентов, объединенных в единое мехатронное устройство, имитирующее участок сборки деталей.

Модуль содержит следующие устройства детектирования и идентификации:

1. Датчик втянутого положения штока цилиндра узла выдачи заготовок – магнитный концевой выключатель В1 типа – D-A93 (характеристики представлены в техническом описании стенда);
2. Датчик выдвинутого положения штока цилиндра узла выдачи заготовок – магнитный концевой выключатель В2 типа – D-A93 (характеристики представлены в техническом описании стенда);

3. Датчик наличия заготовок в магазине узла выдачи заготовок – световой барьер BQ1, типа – BF3RX-P (характеристики представлены в техническом описании стенда);

4. Датчик типа материала заготовки – индуктивный концевой выключатель B3 типа – ВБИ-М18-34У-1111-3 (характеристики представлены в техническом описании стенда);

5. Датчик наличия заготовки в зоне выдачи крышек – световой барьер BQ2, типа – BF3RX-P (характеристики представлены в техническом описании стенда);

6. Датчик наличия заготовки в зоне накопительного узла – диффузионный оптический датчик B4, типа – BEN300-DDT (характеристики представлены в техническом описании стенда);

7. Датчик наличия крышек в магазине узла выдачи крышек 1 – световой барьер BQ3, типа – BF3RX-P (характеристики представлены в техническом описании стенда);

8. Датчик наличия крышек в магазине узла выдачи крышек 2 – световой барьер BQ4, типа – BF3RX-P (характеристики представлены в техническом описании стенда);

Модуль содержит следующие устройства управления:

1. Подача заготовки на ленту из магазина узла выдачи заготовок – пневматический цилиндр двухстороннего действия YA1, типа – CD85N16-60-B, на штоке которого закреплен пластиковый толкатель (характеристики представлены в техническом описании стенда);

Управление пневмоцилиндрами осуществляется с помощью пневмораспределителей с электроуправлением PYA1 типа – SY3120-5LOU-C4-Q, закрепленном на пневмоострове

2. Пуск/стоп двигателя постоянного тока узла конвейера – двигатель постоянного тока с редуктором A1, типа – RB3500090-30Y02R, управляемого контроллером UT-DrDPT-M01 имеющий возможность реверса A2 (характеристики представлены в техническом описании стенда);

3. Комплектовка заготовки крышкой из магазина узла выдачи крышек 1 – пневматический цилиндр двухстороннего действия YA2, типа – CD85N16-60-B, на штоке которого закреплен пластиковый толкатель (характеристики представлены в техническом описании стенда);

Управление пневмоцилиндрами осуществляется с помощью пневмораспределителей с электроуправлением PYA2 типа – SY3120-5LOU-C4-Q, закрепленном на пневмоострове

4. Комплектовка заготовки крышкой из магазина узла выдачи крышек 2 – пневматический цилиндр двухстороннего действия YAZ, типа – CD85N16-60-B, на штоке которого закреплен пластиковый толкатель (характеристики представлены в техническом описании стенда);

Управление пневмоцилиндрами осуществляется с помощью пневмораспределителей с электроуправлением PУАЗ типа – SY3120-5LOU-C4-Q, закрепленном на пневмоострове

Ход работы

Для проведения данной лабораторной работы необходимо использовать мехатронный модуль – сборка деталей (Рисунок 2) и пульт ручного управления модулем (Рисунок 3).

1. Подключить мехатронный модуль (Терминал подключения к ПЛК *UT-CRP-M01*) к Пульту ручного управления;

2. Подавая сигналы с тумблеров пульта ручного управления зафиксировать срабатывание всех устройств управления;

3. Произвести требуемые воздействия для срабатывания устройств детектирования и идентификации и зафиксировать индикацию соответствующих светодиодов пульта симуляции.

4. Выполнить отчет о проделанной работе

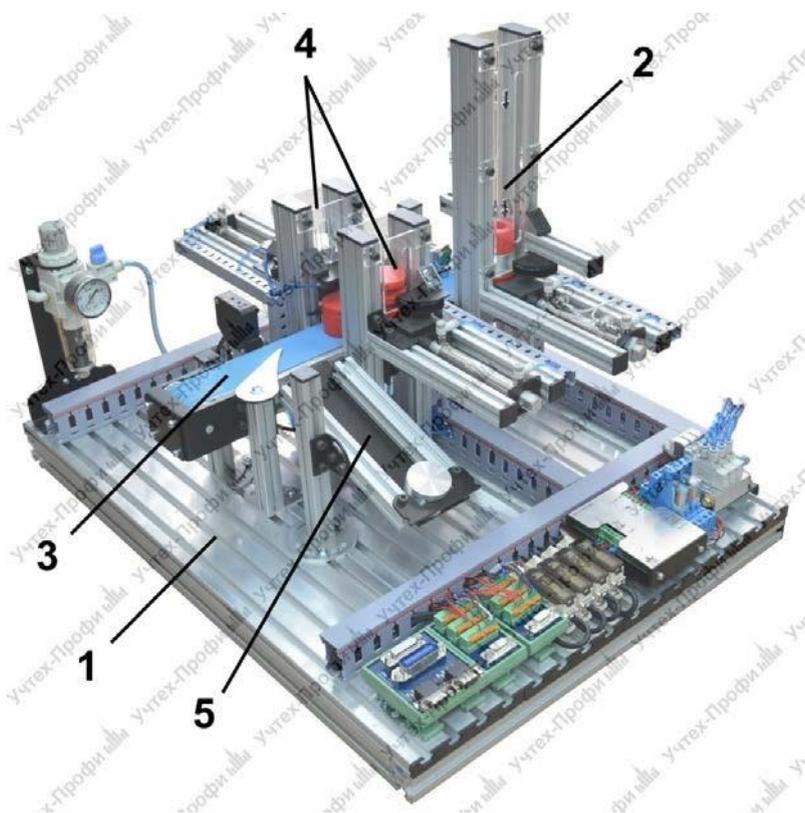


Рисунок 2 – Мехатронный модуль – сборка деталей



Рисунок 3 – Пульт ручного управления модулем

2.2 Работа №2. Изучение дискретной системы управления двигателем постоянного тока на основе программируемого логического контроллера

Цель работы

Ознакомиться с принципом работы дискретной системы управления, изучить назначение и технические характеристики датчиков и устройств управления, приобрести навыки разработки алгоритма управления, а также программирования данной системы с применением специализированного программного обеспечения.

Содержание работы

Дома, при подготовке к работе:

- Изучить назначение, технические характеристики логического моноблока ОВЕН ПЛК160, основные узлы и возможности лабораторного стенда;
- Изучить систему команд и принципы программирования ПЛК160;
- Изучить описание мехатронного модуля, ознакомиться с характеристиками всех его элементов;
- Выполнить синтез системы автоматизации согласно выданному варианту задания;
- Составить программу для ввода в контроллер;

В лаборатории:

- Закрепить полученные при подготовке к лабораторной работе знания на практике;
- Отладить мехатронный модуль в ручном режиме;
- С помощью разработанного алгоритма выполнить управление мехатронной станцией.

Общие сведения

Текущую работу рекомендуется выполнять после успешного завершения лабораторной работы №1 текущих методических указаний, подробное описание устройств детектирования, идентификации и управления переведено в лабораторной работе №1 и в техническом описании стенда.

Для проведения данной лабораторной работы необходимо использовать мехатронный модуль – сборка деталей (Рисунок 4) и лабораторный моноблок «ОВЕН ПЛК 160» (Рисунок 5).

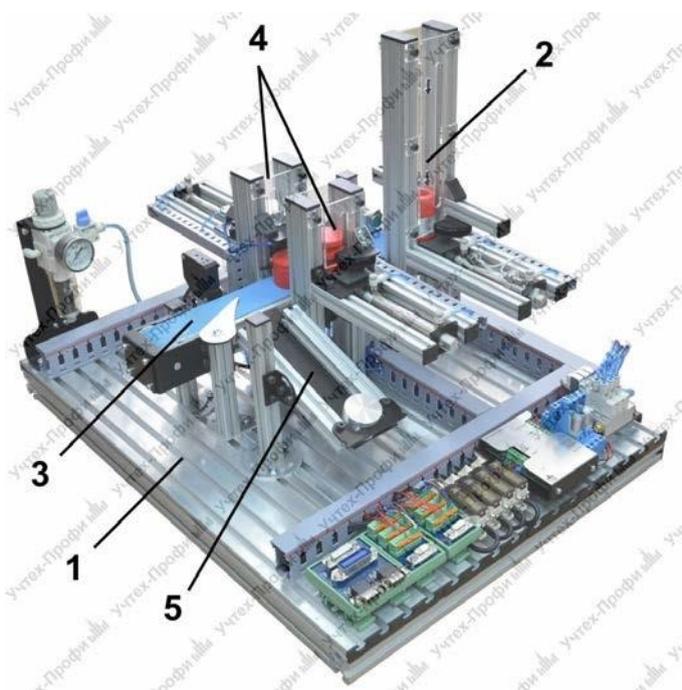


Рисунок 4 – Мехатронный модуль – сборка деталей

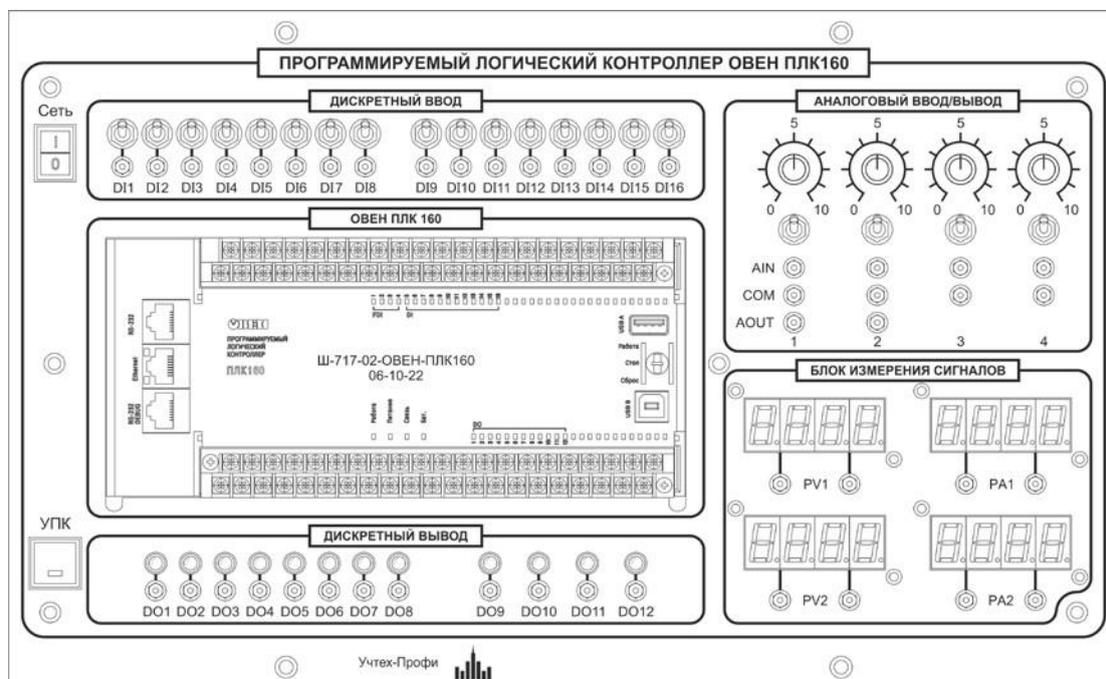


Рисунок 5 – Лабораторный моноблок «ОВЕН ПЛК 160»

Подключить мехатронный модуль (Терминал подключения к ПЛК *UT-CRP-M01*) к ПЛК (разъем X2, который находится на тыльной стороне кожуха моноблока «ОВЕН ПЛК 160»).

При подключении используется кабель 24Pin(M) – 24Pin(F).

В качестве датчика начального положения используется датчик типа материала заготовки – индуктивный концевой выключатель В3 типа – ВБИ-М18-34У-1111-3 (характеристики представлены в техническом описании стенда);

В качестве датчика конечного положения используется датчик наличия заготовки в зоне накопительного узла – диффузионный оптический датчик В4, типа – ВЕН300-DDT (характеристики представлены в техническом описании стенда);

Объект для перемещения – металлическая заготовка.

Ход работы

Выполнить подготовительные работы, установить металлическую деталь в начало конвейера и в соответствии со своим вариантом выполнить индивидуальное задание. Убедиться в правильности работы алгоритма.

Подготовить отчет о выполненной работе.

Таблица 3 – Варианты заданий

	Вариант 1	Вариант 2	Вариант 3	Вариант 4
Пуск автоматического режима управления	DI16	DI15	DI14	DI13
Останов автоматического режима управления	DI13	DI14	DI15	DI16
Индуктивный датчик В3 (правый)	DI4	DI4	DI4	DI4
Диффузионный датчик В4 (левый)	DI6	DI6	DI6	DI6
Пуск двигателя А1	DO2	DO2	DO2	DO2
Реверс двигателя А2	DO3	DO3	DO3	DO3
Задержка t , сек	3	4	2	5
Кол-во циклов k	3	2	4	5

Варианты заданий

При подготовке к лабораторной работе студент должен в соответствии с таблицей (см. Таблица 3) выбрать свой вариант задания и заполнить коммутационную таблицу подключения мехатронного объекта к ПЛК.

Для заданного варианта необходимо разработать алгоритм его решения и составить программу для контроллера ОВЕН ПЛК160.

Запустить программу в работу и убедиться в правильности ее работы. При наличии ошибок в работе устранить их и продемонстрировать преподавателю работу правильно функционирующей программы. Подготовить отчет о выполненной работе.

Вариант 1 и 2.

После включения тумблера «Пуск автоматического режима управления» программа переходит в режим ожидания заготовки правым датчиком. После появления заготовки в поле зрения датчика, лента конвейера перемещает её до левого датчика и ожидает t секунд. Затем конвейер возвращает заготовку в исходное место. Данный цикл повторяется k раз.

Вариант 3 и 4

После включения тумблера «Пуск автоматического режима управления» программа переходит в режим ожидания заготовки правым датчиком. После появления заготовки в поле зрения датчика, лента конвейера перемещает её до левого датчика и сразу же возвращает в исходное место. По истечении t секунд цикл повторяется. Данный цикл повторяется k раз.

2.3 Работа №3. Синтез и анализ алгоритмов управления мехатронной станцией сборки деталей

Цель работы

Ознакомиться с принципом работы дискретной системы управления, изучить назначение и технические характеристики датчиков, приобрести навыки разработки алгоритма управления, а также программирования данной системы с применением специализированного программного обеспечения.

Содержание работы

Дома, при подготовке к работе:

- Изучить назначение, технические характеристики логического моноблока ОВЕН ПЛК160, основные узлы и возможности лабораторного стенда;
- Изучить систему команд и принципы программирования ПЛК160;
- Изучить описание мехатронного модуля, ознакомиться с характеристиками всех его элементов;
- Выполнить синтез системы автоматизации согласно выданному варианту задания;
- Составить программу для ввода в контроллер;

В лаборатории:

- Закрепить полученные при подготовке к лабораторной работе знания на практике;
- Отладить мехатронный модуль в ручном режиме;
- С помощью разработанного алгоритма выполнить управление мехатронной станцией.

Общие сведения

Лабораторную работу №3 рекомендуется выполнять после успешного завершения работ №1 и №2.

Для проведения данной лабораторной работы необходимо использовать лабораторный моноблок «ОВЕН ПЛК 160» (Рисунок 6), мехатронный модуль – сборка деталей (Рисунок 7), пульт ручного управления модулем (Рисунок 8).

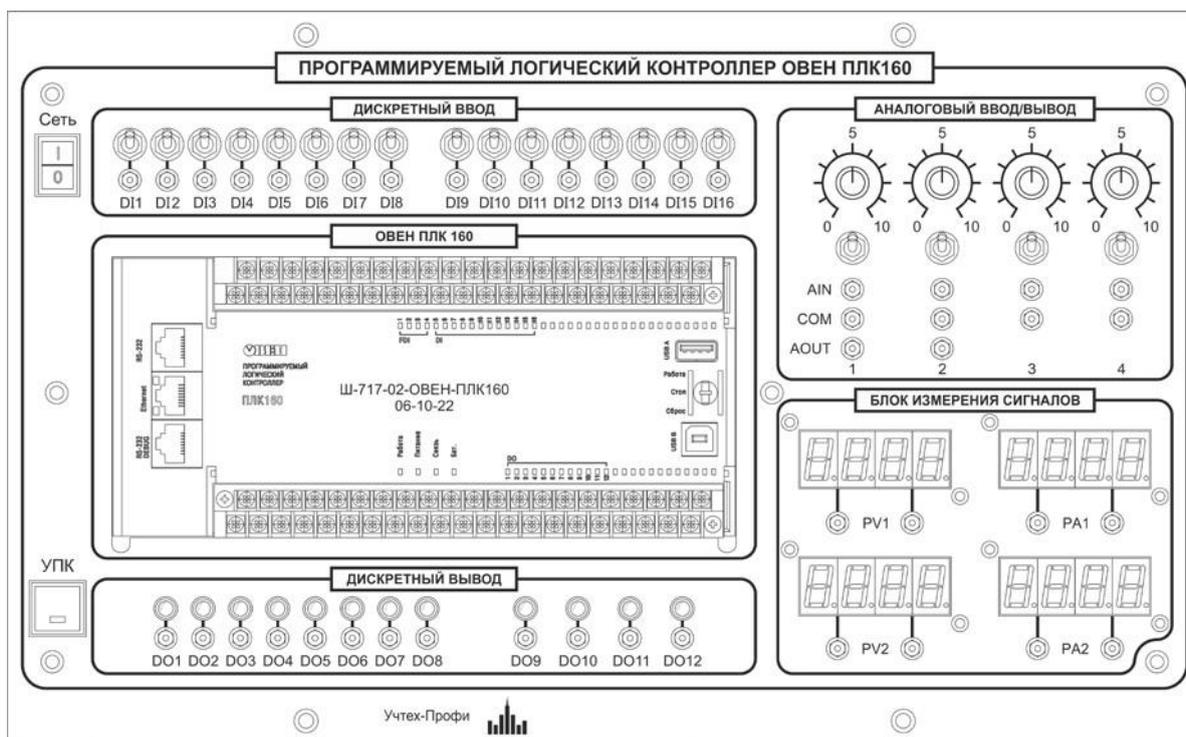


Рисунок 6 – Лабораторный моноблок «ОВЕН ПЛК 160»

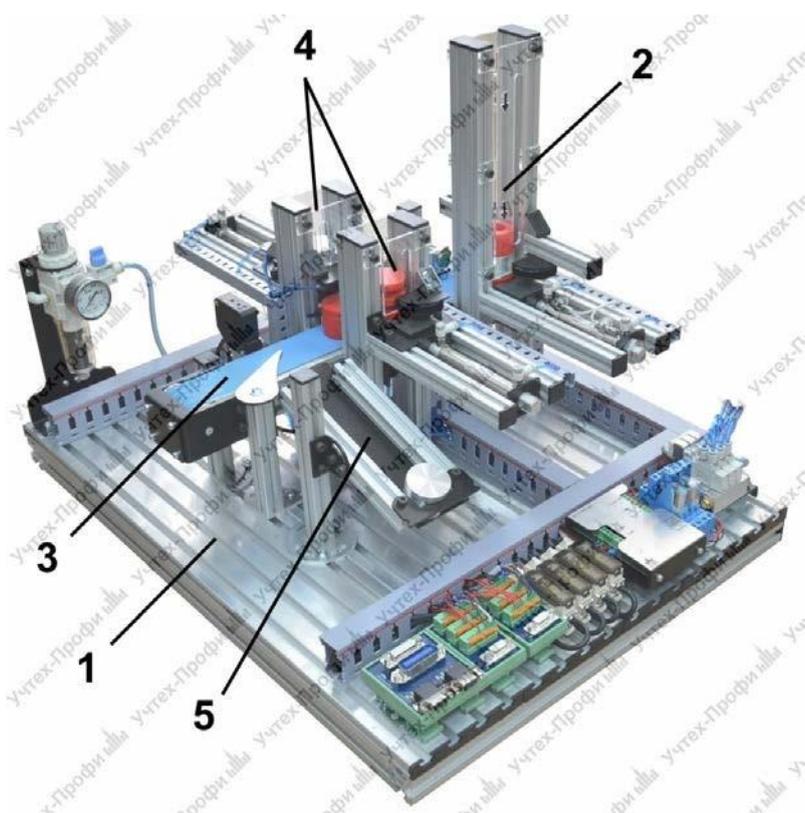


Рисунок 7 – Мехатронный модуль – сборка деталей



Рисунок 8 – Пульт ручного управления модулем

Для подключения мехатронного модуля (Терминал подключения к ПЛК *UT-CRP-M01*) к ПЛК используется разъем X2, который находится на тыльной стороне кожуха моноблока «ОВЕН ПЛК 160».

Для подключения мехатронного модуля (Терминал подключения к ПЛК *UT-CRP-M01*) к Пульту ручного управления используется разъем X1, который находится на тыльной стороне кожуха Пульты ручного управления.

При подключении используется кабель 24Pin(M) – 24Pin(F).

Порядок выполнения работы

Подключить питание к модулям, необходимым для выполнения лабораторной работы. После подключения питания следует выполнить отладку мехатронного модуля в ручном режиме, для этого:

– Подключить мехатронный модуль (Терминал подключения к ПЛК *UT-CRP-M01*) к Пульту ручного управления;

– Подавая сигналы с тумблеров пульта ручного управления (выходные сигналы) выполнить полный цикл в ручном режиме, убедиться в корректной работе датчиков в процессе работы (входные сигналы пульта ручного управления).

Полный цикл работы для мехатронного модуля сборка деталей: в узле выдачи заготовок находятся детали (металлические и пластмассовые красные/синие). При выдачи на конвейер металлической детали и при наличии крышки в узле накопления крышек 1 обеспечить выдачу крышки на деталь и ее последующий спуск в узел накопления деталей. При выдачи на конвейер

пластмассовой детали и при наличии крышки в узле накопления крышек 2 обеспечить выдачу крышки на деталь и ее последующий спуск в узел накопления деталей. При отсутствии необходимой крышки для детали – выполнить перемещение детали в узел накопления. Таблица датчиков и исполнительных элементов приведена в Приложении А технического описания стенда.

При успешном завершении полного цикла составить таблицу подключения устройств к терминалу подключения к ПЛК UT-CRP-M01 (Таблица 4).

Таблица 4 – Пример таблицы соответствия входных/выходных сигналов

Адрес	Комментарий
DI1	Датчик втянутого положения штока цилиндра УВЗ
....
DO1	Подача заготовки на ленту из магазина УВЗ
....

После проведения отладки мехатронного модуля в ручном режиме можно приступать к разработке алгоритма управления мехатронного модуля.

!В качестве примера будет рассмотрен процесс создания алгоритма управления ленточным конвейером. Разработка алгоритма для мехатронной станции – сборка деталей является самостоятельной задачей для подгруппы. (Вариации исполнения цикла работы уточняются у преподавателя).

Разработка алгоритма управления

Разработка алгоритма управления разделяется на следующие этапы:

- Словесное описание полного цикла работы;
- Разработка функциональной схемы процесса;
- Составление таблицы соответствия адресов;
- Составление логических уравнений;
- Составление программы на одном из языков программирования;
- Отладка и проверка проекта на работоспособность.

Словесное описание полного цикла работы

Конвейер начинает работать при появлении детали в области срабатывания щелевого оптического датчика. В зависимости от материала изделия следует

выполнить следующую сортировку – металлические детали поместить в приемный лоток с помощью электрического стоппера, пластмассовые детали – перемещаются до конца конвейера. Определение металлических деталей происходит с помощью, установленного вначале конвейера, индуктивного датчика.

Разработка функциональной схемы процесса

Функциональная схема системы представлена ниже (см. Рисунок 9).

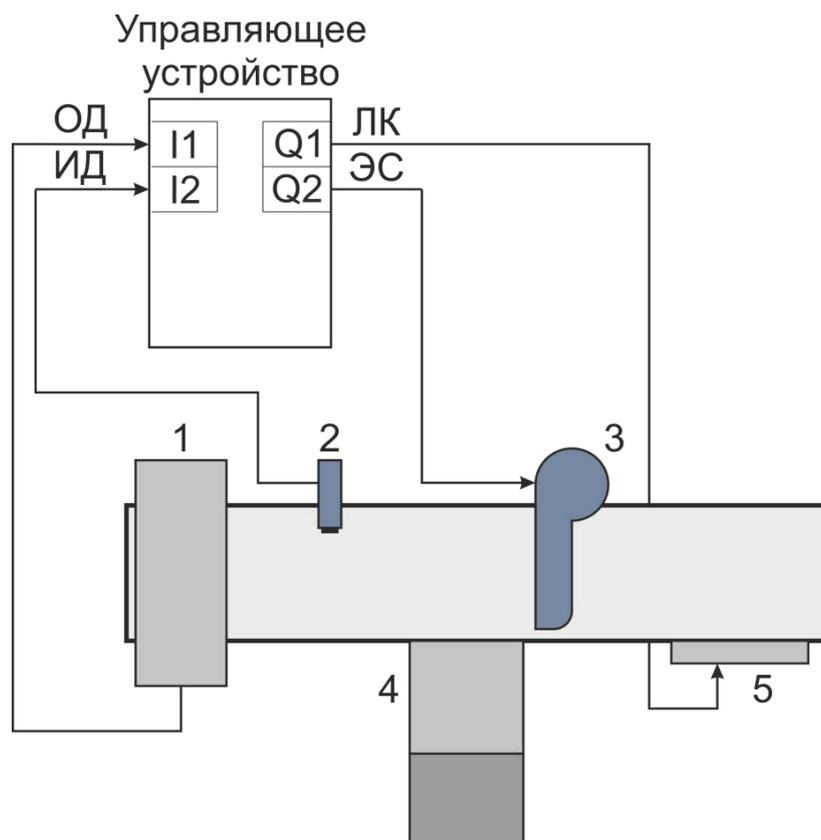


Рисунок 9 – Функциональная схема системы

На функциональной схеме изображается управляющее устройство и его выходные/выходные сигналы. Так же на схеме необходимо отобразить главные узлы технологического процесса, в данном случае: 1 – целевой оптический датчик, 2 – индуктивный датчик, 3 – электрический стоппер, 4 – приемный лоток, 5 – двигатель ленточного конвейера.

Составление таблицы соответствия адресов

Для структурирования и удобства создания программы управления, необходимо составить таблицу соответствия адресов (Таблица 12).

Таблица 5 – Таблица соответствия адресов

Адрес переменной	Имя переменной	Комментарий
I1	ОД	Сигнал с щелевого оптического датчика о наличии заготовки
I2	ИД	Сигнал с индуктивного датчика о наличии металлического объекта на конвейере
Q1	ЛК	Управляющий сигнал на двигатель ленточного конвейера
Q2	ЭС	Управляющий сигнал на электрический стоппер

Составление логических уравнений

Для реализации системы управления технологическим процессом необходимо составить логические уравнения в адресах или в буквенных обозначениях переменных (по усмотрению преподавателя) которые обеспечат корректное выполнение полного цикла работы. В примере будут приведены уравнения в адресах управляющего устройства.

1. При наличии детали в области срабатывания щелевого оптического датчика конвейер начинает свое движение в течении следующих девяти секунд

$$Q1 = (I1 + Q1) \cdot \overline{Q1}^{t9c}$$

↓ – Задержка сигнала по заднему фронту, ↑ – по переднему.

2. При наличии сигнала с индуктивного датчика подается управляющий сигнал на электрический стоппер для помещения металлической детали в накопительный лоток.

$$Q2 = (I2 + Q2) \cdot \overline{Q2}^{t9c}$$

Составление программы на одном из языков программирования

После написания логических уравнений следует создать проект в программной среде CODESYS для ПЛК160. Подробное описание создания проекта представлено в методических указаниях часть 1 и написать программу для последующей загрузки ее в контроллер. Программа написанная на языке ST, для загрузки в управляющее устройство имеет следующий вид (Рисунок 10).

```
PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003     T1: TON;
0004     T2: TON;
0005 END_VAR
0006
0007 <
0001 Q1:=(I1 OR Q1) AND NOT T1.Q; (*Запуск конвейера на 9с*)
0002 T1(IN:=Q1, PT:=T#9S);
0003 Q2:=(I2 OR Q2) AND NOT T2.Q; (*Пуск электрического стоппера на 9с*)
0004 T2(IN:=Q2, PT:=T#9S);
0005
```

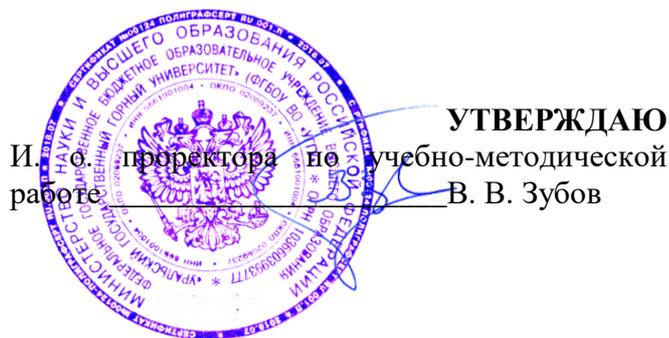
Рисунок 10 – Алгоритм на языке ST для реализации требуемого цикла работы

Отладка и проверка проекта на работоспособность

Последний этап разработки проекта автоматизации – отладка и проверка проекта на работоспособность. Провести тестирование системы в нескольких циклах, убедиться в её корректной работе, при наличии ошибок – провести корректировку проекта.

Задание

Подробно ознакомиться с разобранным примером создания алгоритма управления мехатронным объектом. Выполнить все этапы разработки алгоритма автоматизации технологического процесса для мехатронной станции – сборка деталей. Написать отчёт о проделанной работе.



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Б1.В.09.08 МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В МЕХАТРОНИКЕ И РОБОТОТЕХНИКЕ

Специальность -

21.05.04 Горное дело

Специализация -

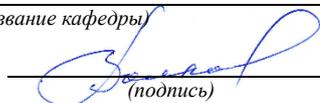
Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой


(подпись)

Волков Е. Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

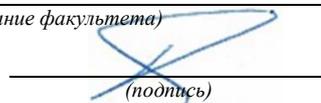
(Дата)

Рассмотрены методической комиссией
факультета

Горно-механического

(название факультета)

Председатель


(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 1 от 18.10.2024

(Дата)

ЕКАТЕРИНБУРГ

ЛЕКЦИЯ 1.	4
ВВЕДЕНИЕ. ОСНОВНЫЕ ТИПЫ БОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ ДЛЯ МИКРОПРОЦЕССОРНЫХ КОМПЛЕКТОВ.	4
Микропроцессоры и микроконтроллеры	5
Микросхемы памяти	6
ЛЕКЦИЯ 2.	17
Программируемые логические интегральные схемы	17
Направление развития микроэлектронных компонентов вычислительных систем	21
ЛЕКЦИЯ 3.	21
АРХИТЕКТУРА И СТРУКТУРА МИКРОПРОЦЕССОРОВ. ЦИКЛ РАБОТЫ МИКРОПРОЦЕССОРА: ТИПЫ АДРЕСАЦИИ И СИСТЕМА КОМАНД.	21
Типы команд микропроцессоров.	21
Структурный параллелизм микропроцессоров	22
ЛЕКЦИЯ 4.	24
Структурные методы уменьшения времени доступа к памяти	24
CISC- и RISC- процессоры.	30
ЛЕКЦИЯ 5.	35
Методы ускорения переключения контекста процессора	35
Стандартизация архитектур микропроцессоров	36
ЛЕКЦИЯ 6.	38
ILP ПРОЦЕССОРЫ	38
Архитектура ILP- процессоров	38
Суперскалярные процессоры	40
VLIW-процессоры.	47
EPIC-процессоры.	48
Направления развития ILP архитектуры.	49
ЛЕКЦИЯ 7.	50
МУЛЬТИСКАЛЯРНЫЕ ПРОЦЕССОРЫ	50
Мультискалярная модель выполнения программы.	50
Мультискалярные программы.	54
Мультискалярные аппаратные средства.	56
Преимущества мультискалярной архитектуры.	57
ЛЕКЦИЯ 8.	58
ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА МП УСТРОЙСТВ С ВНЕШНИМИ УСТРОЙСТВАМИ И ПАМЯТЬЮ. ВСТРОЕННЫЕ МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ.	58
ЛЕКЦИЯ 9.	62
ОДНОКРИСТАЛЬНЫЕ МИКРО-ЭВМ.	62
Общие сведения об однокристальных микроЭВМ семейства МК48.	63
Однокристальные микро-ЭВМ семейства МК51.	65
Структура микроконтроллеров семейства MCS-x96.	74
Периферийные устройства микроконтроллеров семейства MCS-x96.	76
ЛЕКЦИЯ 10.	78
Преимущества регистр-регистровой архитектуры	78
Блок-схема микроконтроллера MCS-196.	80
Регистровый файл	83
Особенность организация памяти микроконтроллеров MCS-196	84
Возможности подключения внешней памяти и внешних устройств ввода/вывода	86
ЛЕКЦИЯ 11.	87
ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРОВ В ПРИВОДАХ МЕХАТРОННЫХ СИСТЕМ. ПРИМЕРЫ МП ПРИВОДОВ СОВРЕМЕННЫХ СИСТЕМ.	87
Пример МК-системы на основе ОМЭВМ семейства МК51.	87

Унифицированная структура силовой части статических преобразователей частоты для широкого класса приводов переменного тока.	89
Универсальный контроллер для встроенных применений	93
Выводы	97
ЛЕКЦИЯ 12.	98
МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ. ТРАНСПЬЮТЕРЫ - ЭЛЕМЕНТНАЯ БАЗА МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ	98
Основные особенности транспьютеров	98
Архитектура и структура транспьютеров фирмы Inmos	99
ЛЕКЦИЯ 13.	109
Транспьютер T9000	109
Транспьютероподобные микропроцессоры серии "Квант"	110

Лекция 1.

Введение. Основные типы больших интегральных схем для микропроцессорных комплектов.

Особая роль в совершенствовании систем автоматического управления в настоящее время принадлежит микроэлектронике и микропроцессорной технике, использование которых в САУ обеспечивает не только улучшение точностных характеристик, повышение отказоустойчивости, обеспечение стабильности функционирования, но и придает САУ принципиально новых свойств, таких, как гибкость и перестраиваемость структуры, адаптивность, способность решать вычислительные и логические задачи, самоконтроль и т.д.

Использование микропроцессоров в САУ требует решения целого ряда задач, специфика которых обусловлена как распределенным управлением в реальном масштабе времени, так и цифровым характером обрабатываемой информации. В связи с этим требуется подготовка специалистов обладающими знаниями и опытом по разработке и эксплуатации микропроцессорных систем различного назначения: информационных, связанных, вычислительных, управляющих и т.д.

Изучение такой интенсивно развивающейся и наукоемкой предметной области, как микроэлектроника, и микропроцессорная техника в частности - задача весьма интересная и сложная, требующая постоянного совершенствования и пополнения получаемых знаний и знакомства со смежными научно-техническими областями. Для эффективного решения прикладных задач любой современный специалист, профессионально связанный с вычислительной техникой, должен иметь адекватное представление о состоянии и перспективах развития ее элементной базы.

Создание фирмой Intel первого микропроцессора в 1971 году положило начало эпохе компьютеризации. "Благодаря микропроцессорам компьютеры стали массовым, общедоступным продуктом", - заявил Тед Хофф (Ted Hoff), один из изобретателей первого микропроцессора. Его имя, вместе с именами его коллег - Федерико Феджина (Federico Faggin) и Стена Мейзора (Stan Mazor), внесено в список лауреатов Национального зала славы изобретателей США, а само изобретение признано одним из величайших достижений XX века.

За чуть более чем четверть вековую историю микропроцессоры прошли поистине гигантский путь. Первый чип Intel 4004 работал на частоте 750 КГц, содержал 2300 транзисторов и стоил около \$200. Производительность его оценивалась в 60 тыс. операций в секунду. На сегодняшний день рекордные показатели принадлежат микропроцессорам Pentium 4 и составляют: 3,46 ГГц, 55 млн. транзисторов, и около \$650 соответственно.

Такое интенсивное развитие технологий в обществе, где основным предметом труда становится информация, является следствием растущего спроса на новые орудия труда - компьютеры. На сегодняшний день компьютеризация является одним из главных направлений научно-

технического прогресса и концентрированным его выражением. Количество и качество производимых в стране компьютеров, степень насыщенности вычислительной техникой самых разных отраслей становится одним из основных критериев ее экономического и военного потенциала.

В формируемом ежегодно в США группой экспертов перечне "критических технологий", охватывающем практически все направления производства, исследований и разработок, оказывающих влияние на военный и экономический статус страны, микроэлектронные технологии традиционно занимают первое место.

В микропроцессорах - наиболее сложных микроэлектронных устройствах - воплощены самые передовые достижения инженерной мысли. В условиях свойственной данной отрасли производства жесткой конкуренции и огромных капиталовложений, выпуск каждой новой модели микропроцессора - так или иначе связан с очередным научным, конструкторским, технологическим прорывом.

Современная элементная база - сверхбольшие интегральные схемы (СБИС), характеризуется большим числом транзисторов на кристалле и относительно малым числом выводов у корпуса. Поэтому БИС адекватны построению логически законченных устройств.

Различные выполняемые функции и сферы применения обусловили специализацию СБИС. Достаточно условно их можно разделить на следующие классы:

1) СБИС с аппаратной реализацией алгоритмов обработки данных:

микропроцессоры универсальные и сигнальные, а также микроконтроллеры, включая интерфейсные схемы для образования мультипроцессорных систем;

2) микросхемы памяти: статические и динамические;

3) программируемые логические интегральные схемы (ПЛИС).

Микропроцессоры и микроконтроллеры

Универсальные микропроцессоры предназначены для применения в вычислительных системах: персональных ЭВМ, рабочих станциях, а в последнее время и в массово-параллельных супер-ЭВМ. Основной их характеристикой является наличие развитых устройств для эффективной реализации операций с плавающей точкой над 64 разрядными и более длинными операндами. Предназначаются в основном для проведения научно-технических расчетов.

Цифровые сигнальные процессоры рассчитаны на обработку в реальном времени цифровых потоков, образованных путем оцифровывания аналоговых сигналов. Это обуславливает их сравнительно малую разрядность и преимущественно целочисленную обработку. Однако современные сигнальные процессоры способны проводить вычисления с плавающей точкой над 32 — 40-разрядными операндами. Кроме того, появился класс медийных процессоров,

представляющих собой законченные системы для обработки аудио- и видеоинформации.

Наибольшей специализацией и разнообразием функций обладают микроконтроллеры, используемые во встроенных системах управления, в том числе в бытовых приборах. Общее число типов кристаллов с различными системами команд превышает 500, и все они, в силу существования изделий с их использованием, имеют свою устойчивую долю рынка.

По прогнозам компаний-производителей дальнейшее развитие технологии производства микропроцессоров будет идти в направлении увеличения плотности транзисторов на кристалле, роста числа слоев металлизации и повышения тактовой частоты, наряду с уменьшением напряжения питания и удельной (на один транзистор) потребляемой электрической и выделяемой тепловой энергии.

Технологический предел линейных размеров транзисторов на кристалле, обусловленный физическими ограничениями, составляет около 0,05 мкм. На пути дальнейшей миниатюризации кроме физических ограничений имеются и экономические. Для каждого следующего поколения микросхем стоимость технологии удваивается (в 1986 г. Intel 80386 имел 250 тыс. транзисторов и выпускался на заводе стоимостью 200 млн. долларов. Завод, производящий микросхемы по технологии 0,25 мкм, стоил 10 млрд. долларов). Возрастают сроки изготовления микропроцессоров, так процессор Pentium производился за шесть месяцев, а более новый Pentium Pro - за девять. Во многом уже сейчас уровень технологии, используемой в массовом производстве, определяется экономическими соображениями. Увеличение числа слоев металлизации экспоненциально повышает процент брака при производстве, увеличение площади кристалла также приводит к снижению выхода годных кристаллов. Современный уровень технологии СБИС позволяет уверенно производить схемы с проектными нормами 0,13-0,09 мкм, 5-6 слоями металлизации, площадью кристалла менее 150 кв.мм и числом транзисторов порядка 55 млн. Смена поколений микропроцессоров происходит каждые 2 года. С каждым поколением линейные размеры элементов уменьшаются в 0,7 раза. В настоящее время ширина проводников составляет 0,13 мкм, а к 2004 г. достигнет 0,09 мкм. Тактовая частота будет составлять более 4 ГГц, что существенно усложнит проблему внутрикристалльных, а еще более межкристалльных соединений.

Основной технологической проблемой будет, по-видимому, отвод тепла от процессоров. Уже сейчас рассеиваемая мощность некоторых процессоров составляет до 70 Вт. И в дальнейшем, с увеличением тактовой частоты и числа транзисторов, ее рост будет продолжаться. Для снижения потребляемой энергии применяются разные технологические способы - уменьшение проектных норм и снижение питающего напряжения. Если мощность и впрямь будет расти, отвод тепла действительно станет непреодолимой преградой.

Микросхемы памяти

Как известно, производительность системы определяется скоростью работы

наиболее «медленного» её компонента. Поэтому для достижения оптимального быстродействия необходимо использовать сбалансированные конфигурации. Это означает, что наиболее важные узлы микропроцессорного устройства, к которым относится процессор, набор системной логики и подсистема памяти, должны обладать приблизительно равной производительностью.

Упрощённо работу микропроцессорного устройства можно представить как последовательность операций, совершаемых над различными данными. Задача процессора заключается в выполнении этих операций, а подсистемы памяти – в доставке данных и сохранении результатов. Взаимодействие процессора и памяти управляется чипсетом либо осуществляется напрямую.

Естественно, что для создания быстрого микропроцессорного устройства необходимо прежде всего обеспечить высокую скорость выполнения операций. Но даже самый совершенный процессор, работая в связке с памятью обладающей недостаточной пропускной способностью, окажется бесполезен, т.к. попросту будет простаивать в ожидании очередных порций информации. Поэтому для современных приложений большое значение имеет пропускная способность памяти, объём и сколько задержки в выдаче данных при каждом обращении (латентность).

Любой подсистеме памяти соответствует набор характеристик, которые более-менее полно будут описывать её возможности. К ним, в первую очередь относится частота работы памяти (частота передачи данных) и разрядность шины. Эти два показателя определяют теоретическую пропускную способность шины памяти, которая равна произведению разрядности и частоты. В действительности теоретическая пропускная способность не достижима из-за присутствия латентности (в среднем 60-80%).

В спецификациях модулей часто указывается время доступа (численно оно равно обратной величине рабочей частоты) и представляет собой длительность одного такта. Т.о., чем меньше время доступа, тем на большей частоте сможет работать модуль памяти.

В настоящее время в микропроцессорных устройствах широко применяется многоуровневая иерархическая архитектура памяти, где на верхнем уровне иерархии расположена более быстродействующая кэш-память, в которую в процессе вычислений помещаются интенсивно используемые фрагменты программного кода и обрабатываемых данных. В реальных системах используется до 3 уровней кэш-памяти. Разработчики используют более быстродействующую и, соответственно, более дорогую память в наиболее "ответственных" узлах системы.

Оперативная память (энергозависимая, т.е. теряющая запомненную информацию при снятии питания) случайного доступа RAM развивается уже довольно долгое время, и к настоящему моменту сформировались две основные её разновидности – статическая (SRAM) и динамическая (DRAM). Их отличие заключается в способе хранения данных. Динамическая память (память использует для запоминания бита информации состояния конденсатора "заряжен - не заряжен") требует постоянного обновления записанной

информации с периодом порядка десятков мсек, что сказывается на увеличении латентности. При этом она проста и дешёва в производстве. Статическая память, наоборот, достаточно дорога за счёт более сложного устройства ячеек (статические микросхемы памяти запоминают биты данных в триггерах, что требует 4-6 транзисторов для хранения бита), но обладает очень низкой латентностью. Тем не менее именно стоимость во многом определяет популярность продукции. Поэтому SRAM не получила большого распространения в модулях ОЗУ, зато активно применяется в процессорах в качестве кеша 2-го уровня.

В одной из первых разновидностей DRAM под названием EDO RAM применялся асинхронный режим доступа (процессор при считывании данных приостанавливает свою работу), что приводило к существенной потере производительности системы. Но несколько лет назад рынок стал заполняться новым типом DRAM – SDRAM с синхронным режимом работы.

Память SDRAM представляет собой набор элементарных ячеек, в каждой из которых может быть записан 1 бит информации. Они объединяются в матрицы, по номерам строк и столбцов которых производится адресация. Физически каждая ячейка DRAM состоит из конденсатора и транзистора. И в этом кроется основной недостаток DRAM. Дело в том, что конденсатор не может самостоятельно поддерживать электрический заряд, поэтому для функционирования ячейку необходимо периодически подзаряжать. Эта процедура должна повторяться и после каждой операции чтения, что в определенных случаях существенно увеличивает латентность.

Поиск ячейки в матрице осуществляется с помощью двух сигналов – RAS и CAS. Первый несёт информацию о номере строки, второй – о номере столбца.

Операцию считывания данных из ячейки можно разбить на несколько элементарных стадий:

пересылка запроса процессора набору системной логики (1 такт системной шины);

перенаправление запроса на банк памяти, соответствующий запрашиваемому адресу (1 цикл работы ОЗУ);

поиск необходимой строки в матрице (2-3 цикла) – “RAS-to-CAS delay” или “RAS-to-CAS latency”;

поиск необходимого столбца в найденной строке, т.е. ячейке, располагающейся на их пересечении (2-3 цикла) – CAS-латентность;

непосредственно считывание и помещение информации в выходной буфер (1 цикл);

передача считанной информации набору системной логики (1 цикл);

перенаправление её процессору (1 такт системной шины).

Т.о., простая на первый взгляд операция растягивается на 9-11 тактов системной шины и соответственно 45—55 тактов процессора при множителе 5. Но столь большая задержка имеет место лишь при считывании первых 8

байтов. Тогда как другие ячейки из той же строки будут доступны уже по истечении лишь времени CAS-латентности. Если же происходит последовательное чтение ячеек, то задержки будут отсутствовать вовсе. С другой стороны, возможен вариант, когда первая и вторая считываемые ячейки располагаются в одном банке, но на разных строках. В этом случае к выше перечисленным задержкам добавится ещё одна – “Precharge delay”, необходимая на подзарядку первой ячейки после операции считывания (2-3 цикла).

Три выше упомянутых параметра - CAS-латентность, RAS-to-CAS-латентность и “Precharge delay” – образуют набор таймингов модуля памяти.

Общая латентность системы складывается не только из задержек выдачи данных ОЗУ, но и времени их пересылки процессору.

Наименьшие потери при пересылке данных будут иметь место при синхронном обмене информацией, т.е. когда частоты системной шины и шины памяти совпадают. При использовании асинхронного режима появляется дополнительная латентность, вызванная необходимостью согласования двух шин.

DDR SDRAM является эволюционным шагом в развитии обычной памяти SDRAM. Основное отличие между ними заключается в способе передачи данных. В DDR SDRAM используется так называемая DDR (Double Data Rate) технология, при которой для кодирования информации используются как передний, так и задний фронт тактового импульса. В результате при той же частоте FSB эффективная частота шины памяти удваивается. При этом на удвоенной частоте передаются только данные, а пересылка адресов осуществляется на основной частоте. В остальном модули SDRAM и DDR SDRAM мало отличаются друг от друга.

Подобная схема передачи данных весьма чувствительна даже к самой незначительной задержке тактового сигнала, поэтому, в отличие от SDRAM, для синхронизации передачи данных помимо сигнала тактового генератора применяется дополнительный сигнал DQS. Этот сигнал передается параллельно с тактовым сигналом и корректирует процесс передачи данных для каждой линии шины данных. Для выдачи сигнала DQS, означающего доступность данных на выходе, используется специальный цикл с фиксированной задержкой DLL (Delay Locked Loop).

Для точной синхронизации сигналов DQS и передачи данных (DQ) соответствующие проводники имеют одинаковую длину и емкость. Таким образом, колебания напряжения и температуры одинаково отражаются на обоих сигналах, что обеспечивает отсутствие нарушения синхронизации между контроллером памяти и микросхемами модулей памяти. Дополнительный контроль синхронизации передачи данных обеспечивает более надежную и высокую скорость передачи данных, чем при использовании только сигнала тактового генератора. Чтение и запись данных производится не в соответствии с тактовым сигналом, а синхронизируется сигнал DQS (рис. 1).

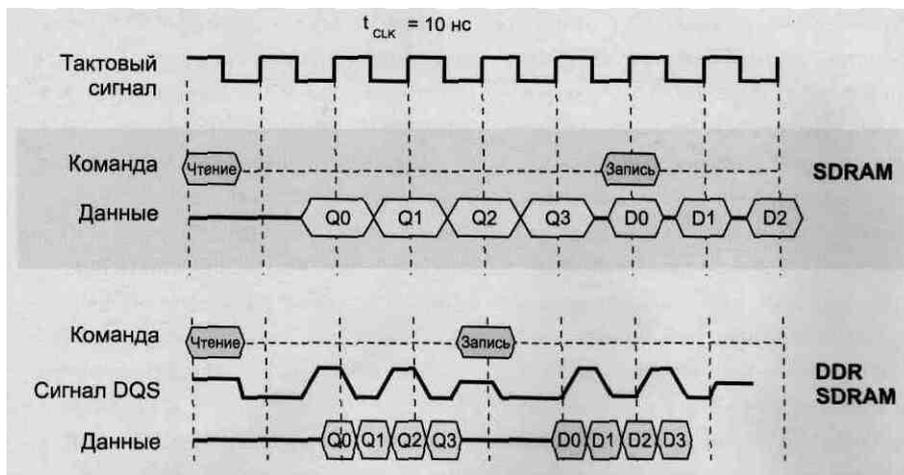


Рис. 1. Временная диаграмма DDR SDRAM

Архитектура микросхемы состоит из четырех независимых банков (рис. 2).

При тактовой частоте 100 и 133 МГц пропускная способность шины памяти DDR SDRAM составляет соответственно 1,6 и 2,1 Гбайт/с.

В настоящее время тактовая частота шины памяти с поддержкой DDR SDRAM составляет 100, 133, 166, 200 МГц, соответствующие микросхемы памяти называются соответственно DDR200, DDR266, DDR333 и DDR400, а модули памяти PC1600, PC2100, PC2700 и PC3200 (в названии модулей указана фактическая пропускная способности шины памяти в Гб/сек).

Дальнейшим развитием технологии DDR SDRAM является технология DDR II. Появление этого типа памяти ожидается в 2003 году. В отличие от DDR SDRAM в DDR II за один такт по каждой линии будет перерываться не 2, а 4 бита информации (при 64-разрядной шине — 32 байта), что вдвое увеличит пропускную способность шины памяти. Естественно, данный подход требует более совершенной системы синхронизации. Рабочие напряжения DDR II уменьшены с 2,5 до 1,8 В. Микросхемы выполнены в корпусе микро-BGA.

Первоначально модули DDR-II будут работать на эффективных частотах 400, 533 и 667 МГц, что позволит говорить о теоретической пропускной способности 3,2, 4,2 и 5,3 Гб/сек.

Для уменьшения времени доступа к данным планируется использовать технологию Virtual Channel (разработанную компанией NEC), а также кэширование информации.

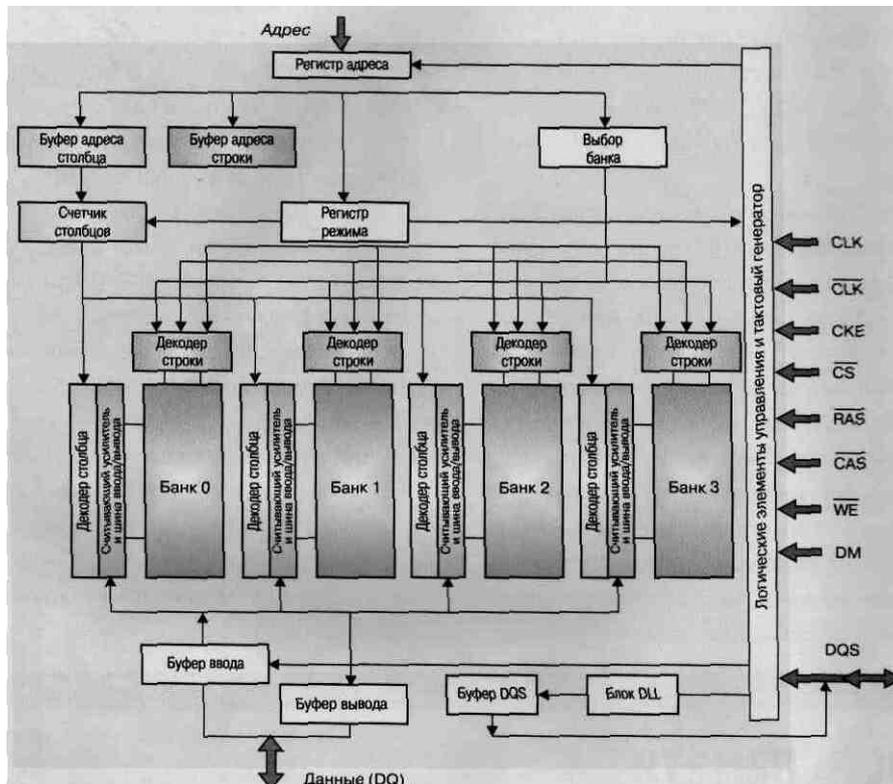


Рис. 2. Структурная схема DDR SDRAM

Следующий большой шаг в развитии DDR SDRAM должен будет произойти лишь в 2007 году, когда планируется появление памяти следующего стандарта DDR-III. По предварительным данным, стартовая эффективная частота модулей составит 800 МГц, которая впоследствии вырастет до 1,5 ГГц. Планируется дальнейшее понижение напряжения питания до 1,2 В. Будут приняты специальные меры по борьбе с возникающими на высоких частотах помехами.

В ближайшее время может появиться ещё один тип ОЗУ – QBM SDRAM. QBM (Quad Band Memory) фактически представляет собой слегка усовершенствованную память DDR, но с увеличенной вдвое теоретической пропускной способностью. При производстве модулей QBM используются те же самые чипы DDR SDRAM. Однако циклы работы половины из них смещены на четверть такта. В результате ширина шины как бы увеличивается до 128 бит, а модуль как бы превращается в двухканальный (для PC2100 пропускная способность составит 4,2 Гб/сек).

Повышение пропускной способности шины памяти может осуществляться либо за счёт увеличения тактовой частоты шины, либо её разрядности. К сожалению, внутреннее устройство SDRAM не позволяет существенно продвинуться в этой области. Дело в том, что каждая разрядная линия шины памяти при распространении по ней переменного сигнала неизбежно излучает электромагнитные волны. Чем выше частота сигнала и плотность расположения линий, тем сильнее влияние излучаемых помех на сигнал. Поэтому у памяти SDRAM существует предел пропускной способности. Чтобы как можно ближе к нему подобраться, можно увеличивать частоту шины при неизменном количестве разрядов. Но есть и альтернативное решение – заметно сократить количество линий в шине, что позволит существенно увеличить

частоты. Именно эта идея и была реализована в памяти Rambus DRAM.

Разрядность шины большинства модулей RDRAM составляет 16 бит, а рабочая частота 400 МГц. К тому же в памяти RDRAM используется DDR-технология. Поэтому эффективная частота шины равняется 800 МГц. Модули RDRAM должны устанавливаться парами, т.к. предназначены для работы в двухканальной конфигурации. Т.о., суммарная теоретическая пропускная способность – 3,2 Гб/сек.

Несмотря на принципиально иной подход к организации памяти, ячейки RDRAM не отличаются от ячеек SDRAM – это та же совокупность транзистора и конденсатора. Тем не менее для реализации RDRAM потребовалось перестроить инфраструктуру модулей. Если у SDRAM один банк состоит из 8 чипов, то у RDRAM, наоборот, каждый чип является независимым и сам может состоять из нескольких десятков банков. В результате такого «распараллеливания» RDRAM способна производить гораздо больше, чем SDRAM, операций чтения за один и тот же промежуток времени. Причём латентность при считывании окажется незначительной как раз из-за того, что многие банки памяти могут постоянно держаться открытыми. С помощью подобных технических решений достигается высокая эффективность использования шины памяти – более 90%.

Обратная сторона медали заключается в усложнении модулей. В отличие от SDRAM чипы RDRAM необходимо оснащать управляющей логикой, что увеличивает их площадь, а высокая частота функционирования приводит к значительному тепловыделению. Более того, сложность в проектировании и производстве определила высокую конечную стоимость, что и сказалось в итоге на отсутствии популярности данного типа памяти среди массового потребителя.

Но у RDRAM есть ещё один серьёзный недостаток – высокая латентность при выдаче первых байтов информации, которая может в 2-3 раза превышать соответствующую латентность памяти SDRAM. В результате RDRAM показывает рекордную пропускную способность при считывании потоковых данных, располагающихся друг за другом. Но в задачах с хаотическим характером обращения к памяти эффективность использования шины памяти резко снижается.

Вся подсистема памяти Rambus состоит из следующих компонентов (рис. 3):

- контроллера памяти (RMC — Rambus Memory Controller);
- Rambus-канала (RC — Rambus Channel);
- генератора тактовых импульсов (DRCG — Direct Rambus Clock Generator);
- микросхемы памяти RDRAM (Rambus DRAM), которая установлена на модулях памяти RIMM (Rambus In-line Memory Module).

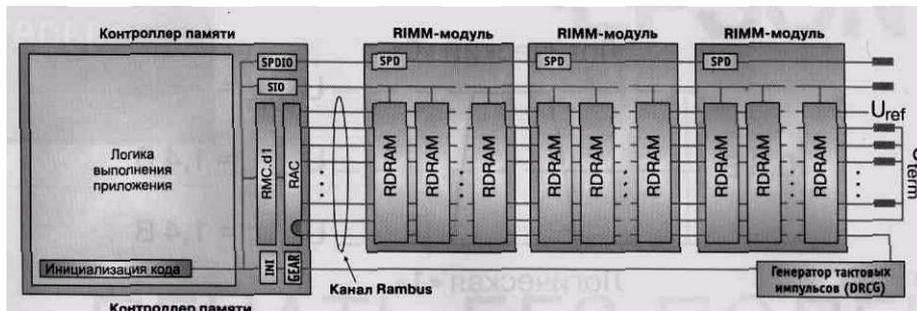


Рис.3 Подсистема памяти RambusDRAM.

Rambus-канал

Rambus-канал (рис. 4) предназначен для осуществления электрической связи между контроллером памяти и микросхемами RDRAM. Физически он представляет собой шину, которая состоит из 30 проводников, разведенных строго параллельно на материнской плате.

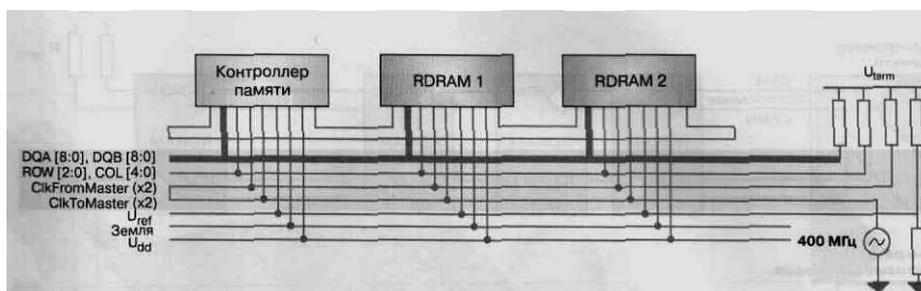


Рис.4 Схема Rambus-канала

Сигналы в Rambus-канале передаются в соответствии с высокоскоростным протоколом сигналов RSL (Rambus Signal Levels). Согласно данному протоколу осуществляется низковольтный перенос номинальных напряжений логического «0» и логической «1» с разностью 800 мВ (рис. 5).



Рис.5

Генератор тактовых импульсов

Rambus-канал является синхронным. Команды и данные передаются параллельно по переднему и заднему фронту синхроимпульсов, создаваемых генератором тактовых импульсов (рис.6).

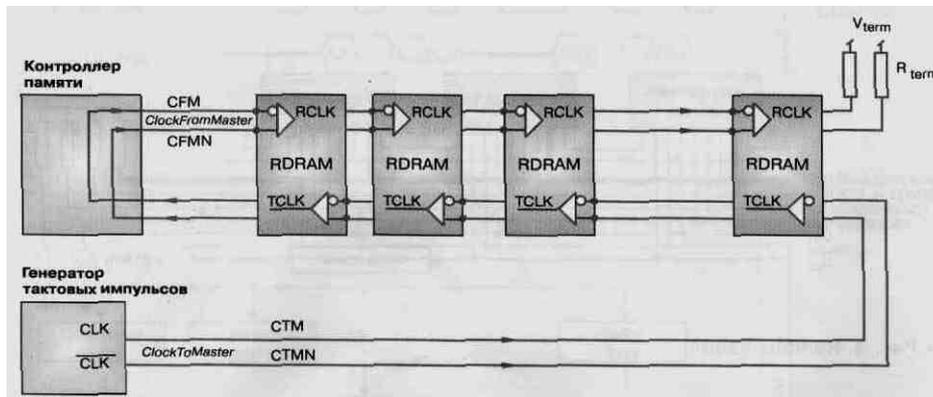


Рис.6 Схема генератора тактовых импульсов

Конструктивно генератор тактовых импульсов может быть интегрирован в контроллер памяти или установлен в виде отдельной микросхемы (в 24-контактном корпусе типа SSOP).

Передача данных

Данные и служебные биты передаются по каждому фронту тактового сигнала с частотой 800 МГц (1,25 нс), что соответствует скорости передачи 800 Мбит/с на линию. Таким образом, пропускная способность шины составляет 1,6 Гбайт/с. Передача данных осуществляется только между контроллером и микросхемами памяти, обмен только между микросхемами невозможен.

Вся информация передается по Rambus-каналу пакетами (пакеты строк, столбцов и данных). Каждый пакет передается за четыре ре такта (10 нс).

Конвейеризация

Операция чтения/записи в Rambus-канале могут быть конвейеризированы. В этом случае время задержки первого пакета составляет 50 нс, далее операции чтения/записи осуществляются практически непрерывно.

При последовательном чередовании двух операций записи и двух операций чтения, адресованных к одной микросхеме, пропускная способность канала составляет 86% от предельной, а при адресации к разным микросхемам — 95%.

Контроллер памяти

Контроллер памяти Rambus осуществляет управление системой памяти RDRAM и основные функции мультиплексирования/демультиплексирования при преобразовании данных, передающихся по 16-разрядной последовательной шине Rambus-канала в 64-разрядную системную шину. Контроллер принадлежит к семейству специализированных интегрированных микросхем ASIC (Application Specific Integrated Circuits). Данные микросхемы способны объединить в себе функции, обычно реализуемые целым набором микросхем, но при этом оказываются существенно быстрее, компактнее и дешевле.

Основным элементом контроллера памяти Rambus (RMC — Rambus Memory Controller) является специализированная микросхема библиотеки макроядра (RAC — Rambus ASIC Cell). Здесь происходит преобразование высокоскоростных сигналов RSL канала Rambus в низкоскоростные сигналы CMOS-уровня.

Контроллер памяти осуществляет взаимодействие подсистемы памяти Rambus с другими компонентами персонального компьютера, взаимодействующими с памятью (центральным процессором, жестким диском, видеосистемой и др.).

Структурная схема микросхемы RDRAM представлена на рис. 7.

Внутреннее ядро RDRAM имеет 128/144-разрядную шину, работающую на частоте 100 МГц - 1/8 тактовой частоты Rambus-канала. Таким образом, каждые 10 нс в ядро (или из ядра) может быть передано 16 байт данных.

Компания Rambus планирует эволюционное развитие существующей линейки модулей RDRAM. В 2003-2004 годах ожидается выпуск как 32-, так и 64-битных чипов, работающих на эффективных частотах 1,2 и 1,33 ГГц (10,6 Гб/сек).

Параллельно Rambus представила новый сигнальный интерфейс памяти Yellowstone. Для реализации данного интерфейса планируется использовать три разработки: дифференциальный протокол передачи запросов, технологию ODR (Octal Data Rate), позволяющую за один такт передавать 8 бит данных, и технологию упрощающую привязку данных к тактовой частоте. По заявлению компании Rambus, технология Yellowstone позволит достичь тактовых частот 6,4 ГГц и пропускной способности до 100 Гб/сек.

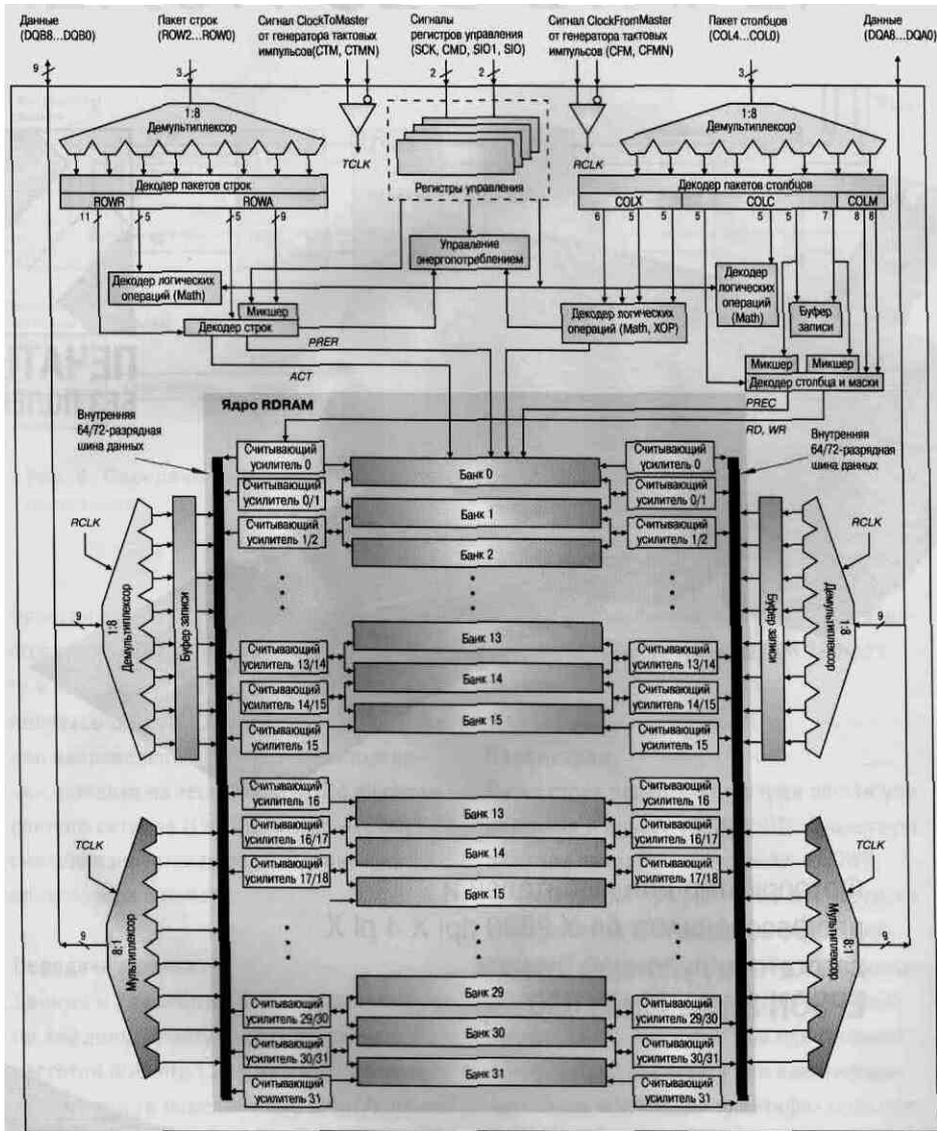


Рис.7 Структурная схема RDRAM

Год	Тип памяти	Частота шины, МГц	Разрядность шины памяти, бит	Максимальная пропускная способность, Мбайт/с	Пропускная способность линии памяти, Мбит/с/контакт
1987	FPM DRAM	25, 33	32	100, 132	24; 32
1995	EDO DRAM	40, 50	32	160, 200	40; 50
1997	PC 66 SDRAM	66	64	528	65,6
1998	PC 100 SDRAM	100	64	800	100
1999/2000	PC 133 SDRAM	133	64	1064	133
1999/2000	Direct RDRAM (один канал)	400, 600, 800*	16	800, 1200, 1600	400; 600; 800
2000	DDR200 SDRAM	200*	64	1600	200
2000	DDR266 SDRAM	266*	64	2100	256
2001	Direct RDRAM (два канала)	400, 600, 800*	16	1600, 2400, 3200	400; 600; 800
2001	Direct RDRAM	1066*	16	2132	1066
2001	DDR300	300*	64	2400	300
2002	DDR333	333*	64	2700	337
2003	DDR II	200; 400*	64	3200, 6400	400; 800

Лекция 2.

Программируемые логические интегральные схемы

ПЛИС - это матричные большие интегральные схемы, позволяющие программно сконструировать в одном корпусе электронную схему, эквивалентную схеме, включающей от нескольких десятков до нескольких сотен ИС стандартной логики. По сравнению с другими микроэлектронными технологиями, в том числе базовыми матричными кристаллами (БМК), технология ПЛИС обеспечивает рекордно короткий проектно-технологический цикл (от нескольких часов до нескольких дней), минимальные затраты на проектирование, максимальную гибкость при необходимости модификации аппаратуры.

ПЛИС содержат программируемую матрицу элементов логического И, программируемую или фиксируемую матрицу элементов логического ИЛИ и так называемые макро ячейки. Макро ячейки как правило включают в себя триггер, тристабильный буфер и вентиль исключающее ИЛИ, управляющий уровнем активности сигнала. Размерность матриц и конфигурация макро ячеек определяют степень интеграции и логическую мощность ПЛИС. В сочетании с разнообразными обратными связями перечисленные элементы формируют завершённую автоматную структуру, ориентированную на реализацию как комбинационных (дешифраторов, мультиплексоров, сумматоров), так и последовательных схем (управляющих автоматов, контроллеров, счётчиков).

В ПЛИС заложены возможности, которые позволяют превратить её в ИС любой функцией цифровой логики. Проектирование сводится к выявлению программируемых элементов (перемычек или запоминающих ячеек), после удаления которых в структуре схемы остаются только связи, которые необходимы для выполнения требуемых функций. На практике эта задача весьма не простая, т.к. современные ПЛИС содержат в среднем несколько десятков тысяч перемычек. Поэтому для проектирования применяют САПР ПЛИС. Каждая компания - производитель ПЛИС разрабатывает и выпускает свою САПР, обеспечивающую реализацию всех этапов проектирования для каждого типа программируемой логики. Благодаря различным САПР, а также структурным и технологическим особенностям, ПЛИС представляют технологию рекордно-короткого цикла разработки РЭА.

Если за рубежом ПЛИС уже заняли заметное место в арсенале разработчика РЭА, то в России эта технология только начинает по-настоящему развиваться.

В каких случаях целесообразно применять ПЛИС?

При разработке оригинальной аппаратуры, а также для замены обычных ИС. При этом значительно уменьшаются размеры устройства, снижается потребляемая мощность и повышается надёжность. (Наиболее эффективно использование ПЛИС в изделиях, требующих нестандартных схемотехнических решений. В этих случаях ПЛИС даже средней степени интеграции (24 вывода) заменяет, как правило, до 10-15 обычных ИС.)

При необходимости резко сократить сроки и затраты на проектирование, а также повысить возможность модификации и отладки аппаратуры. Поэтому ПЛИС широко применяется в стендовом оборудовании, на этапах разработки и производства опытной партии новых изделий, а также для эмуляции схем, подлежащих последующей реализации на другой элементной базе.

Отдельная область применения ПЛИС – проектирование на их основе устройств для защиты программного обеспечения и аппаратуры от несанкционированного доступа и копирования. ПЛИС обладают такой технологической особенностью, как «бит секретности», после программирования которого схема становится недоступной для чтения. Обычно применение 1-2 ПЛИС средней степени интеграции оказывается вполне достаточной для надёжной защиты информации.

Наиболее широко ПЛИС используются в микропроцессорной технике. На их основе разрабатываются контроллеры, адресные дешифраторы, логика обрaмления микропроцессоров, формирователи управляющих сигналов и др. На ПЛИС часто изготавливают микропрограммные автоматы и другие специализированные устройства (цифровые фильтры, схемы обработки сигналов и изображений, процессоры быстрого преобразования Фурье, аппаратура уплотнения телефонных сигналов и т.д.).

Применение ПЛИС становится актуальным ещё и потому, что у разработчика зачастую нет необходимых стандартных микросхем.

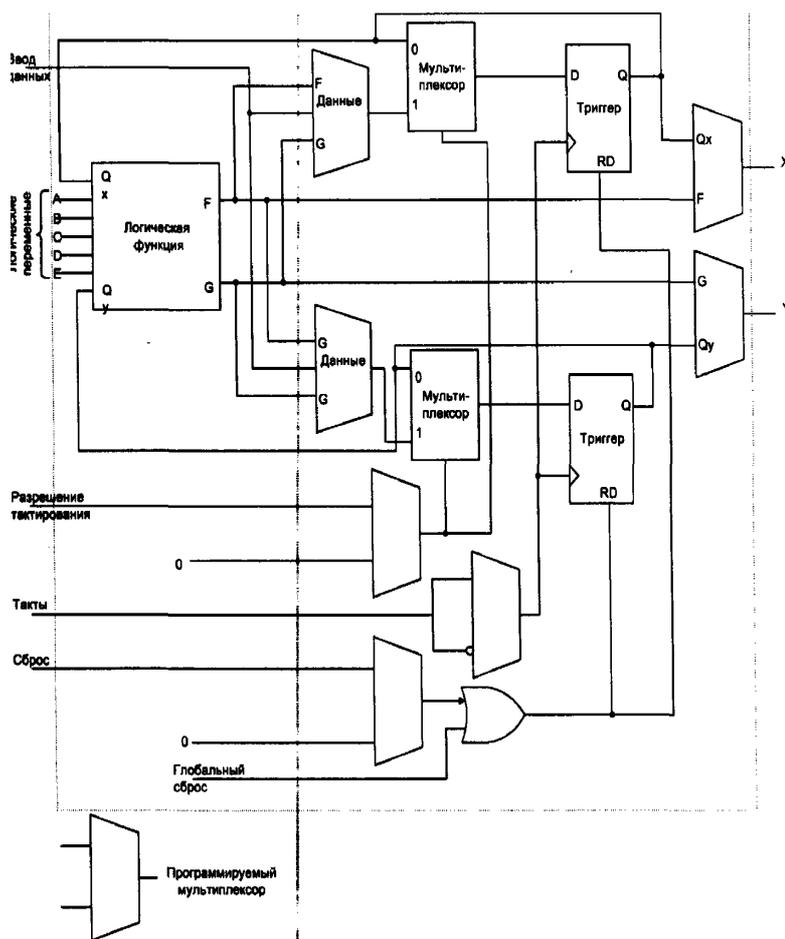


Рис. 8. Логический блок ПЛИС

В настоящее время на мировом рынке можно отметить несколько основных компаний-производителей — ПЛИС-XILINX, ALTERA, LATTICE, AT&T, INTEL, выпускающих микросхемы с архитектурой EPLD (EPROM technology based complex Programmable Logic Device) - многократно программируемые, и FPGA (Field Programmable Gate Array) - многократно реконфигурируемые.

В качестве памяти для хранения конфигурации в ПЛИС EPLD используется ППЗУ с ультрафиолетовым стиранием, а у ПЛИС FPGA - статическое ОЗУ.

Микросхема FPGA представляет собой матрицу логических ячеек, соединенных между собой логическими ключами. Содержащаяся в микросхемах FPGA статическая память, будучи заполненной определенной битовой последовательностью, воздействует на логические ячейки и соединяющие их ключи, позволяя получить требуемые электрические схемы (регистры, счетчики, логические схемы и т.д., соединенные друг с другом в требуемом порядке). Каждая микросхема FPGA имеет также вход для записи битовой последовательности, заполняющей статическую память, а также элементы "вход/выход" для связи с другими микросхемами.

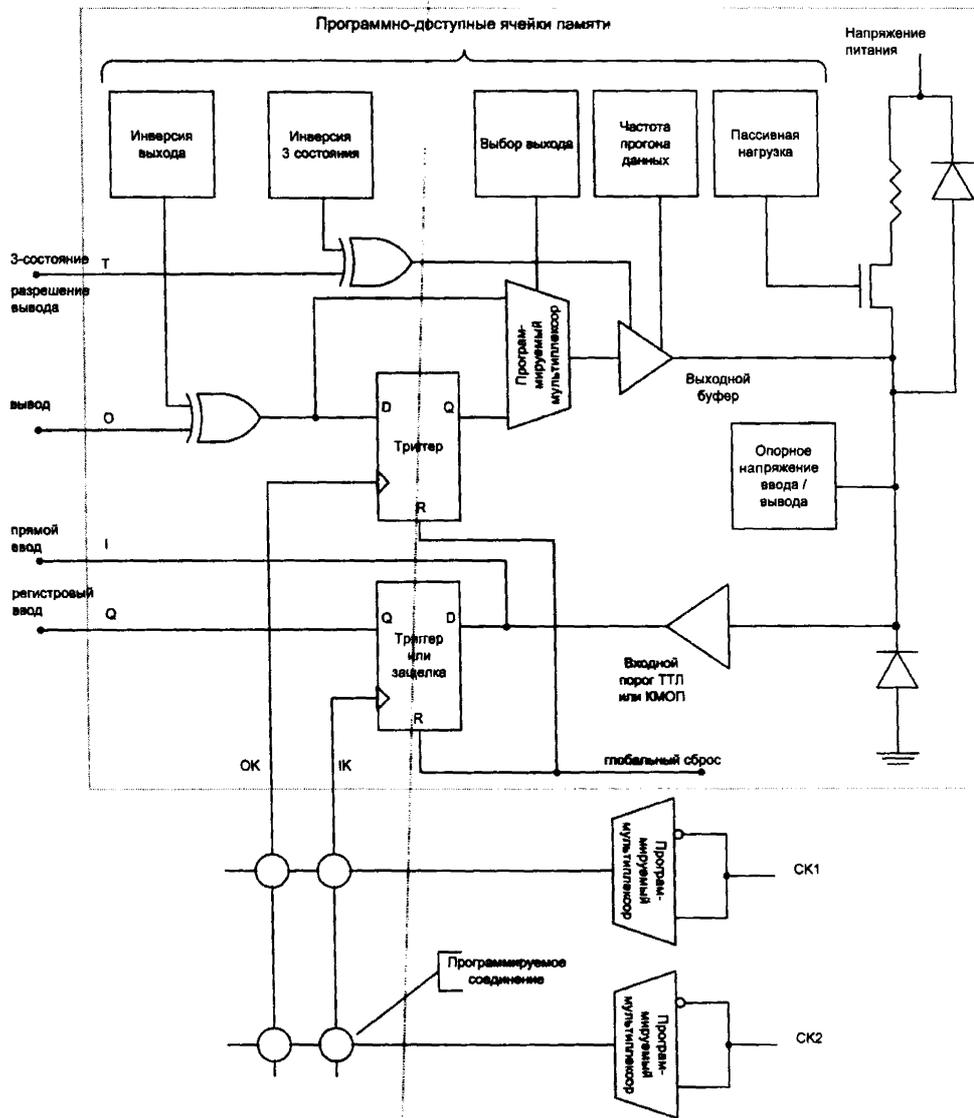


Рис. 9. Блок ввода/вывода ПЛИС

Существует крайняя точка зрения, что со временем ПЛИС вытеснят серийные микропроцессоры, так как пользователь сможет реализовать требуемую ему систему программными средствами, учитывая всю специфику своего приложения.

Таким образом, на основе одной или нескольких микросхем FPGA можно создать реконфигурируемый процессор, обладающий преимуществами спецпроцессора на "жесткой" логике, но способного путем изменения содержимого статической памяти решать любые задачи, подобно универсальному процессору.

Логический блок - один из базовых элементов архитектуры ПЛИС FPGA, может выполнять любую логическую функцию в соответствии с заданной битовой последовательностью (рис.8). Изменять выполняемую функцию можно неограниченное количество раз путем загрузки другой битовой последовательности.

Блок ввода/вывода (рис.9), так же как и логический блок, может быть настроен на выполнение любого электрического соединения реализованной внутри ПЛИС схемы с внешним миром через соответствующий контакт

микросхемы.

Направление развития микроэлектронных компонентов вычислительных систем

Для создания высокоэффективных вычислительных систем важно интегрировать на кристалле как можно больше функций по обработке и хранению данных, а также интерфейс с пользователем и другими вычислительными системами.

Стремление к интеграции совокупности функций обусловлено рядом факторов. Во-первых, при однокристалльной реализации пропускная способность интерфейсов между подсистемами обработки и хранения данных не ограничивается количеством выводов корпуса кристалла и может достигать требуемого значения. Во-вторых, упрощается системная плата, на которую монтируется кристалл, уменьшается объем монтажных работ, повышаются показатели надежности и производительности вычислительной системы и уменьшается ее стоимость. В-третьих, снижаются требования к количеству выводов корпуса кристалла, так как минимизируется интерфейс с другими компонентами вычислительной системы, например с устройствами отображения информации. И, наконец, в четвертых, реализуются возможности миллионов транзисторов, которые могут быть размещены на кристалле. Эти транзисторы можно использовать как для построения специализированной системы, состоящей из совокупности проблемно-ориентированных блоков, так и для создания параллельных систем из нескольких процессоров.

Наряду с созданием однокристалльных систем существует проблема организации быстрых интерфейсов между микросхемами в многокристалльных системах, например между микросхемами процессора и памяти. В Pentium Pro эта проблема решается путем размещения в одном корпусе двух кристаллов: собственно микропроцессора и кэш-памяти второго уровня. Другое возможное решение этой проблемы заключается в создании многокристалльных микросборок, в которых бескорпусные СБИС монтируются на кремниевой подложке с нанесенными, возможно несколькими, слоями межкристалльных соединений.

Лекция 3.

Архитектура и структура микропроцессоров. Цикл работы микропроцессора: типы адресации и система команд.

Типы команд микропроцессоров.

В ходе эволюционного развития архитектур процессоров в состав системы команд вводились и, в силу преемственности программного обеспечения, закреплялись сложные команды, которые по мнению разработчиков соответствовали решаемым задачам. Мерой этого соответствия чаще всего был объем двоичного кода программы, так как минимизация длины программы

была равнозначна минимизации времени исполнения. Команды бывают разных типов: "регистр, регистр -> регистр", "память, память -> память", "регистр -> память" и др. Сложные команды модифицируют содержимое групп регистров и ячеек памяти, и для их реализации при приемлемых затратах оборудования, как правило, применяется микропрограммирование.

Команды называются скалярными, если входные операнды и результат являются числами (скалярами).

Команды называются векторными, если входные операнды и, возможно, результат являются вектором (массивом) чисел, а для преобразования данных массива (вектора) используется одна векторная команда. Примером векторной команды служит команда, при выполнении которой умножаются два очередных элемента двух массивов, далее произведение суммируется с содержимым некоторого заданного регистра, после чего модифицируются адреса памяти для доступа к двум очередным элементам массивов. Указанная последовательность действий повторяется заданное число раз по счетчику, определенному в теле команды.

Само появление векторных команд обусловлено стремлением ускорить обработку массивов данных за счет исключения затрат времени на выборку и дешифрацию команд обработки, одинаковых для всех компонент входных массивов.

Однако использование векторных команд требует подготовки программистом векторизованного кода программ, что, вообще говоря, эквивалентно разработке параллельных программ.

При сохранении последовательных программ для ускорения обработки применяются суперскалярные процессоры, в которых за счет параллельной работы функциональных устройств процессора в одном такте вырабатывается несколько скалярных результатов.

Структурный параллелизм микропроцессоров

Повышение производительности микропроцессоров достигается за счет увеличения тактовой частоты, совершенствования параллельной и конвейерной обработки данных, а также уменьшения времени доступа к памяти. Современные микропроцессоры содержат десять и более обрабатывающих устройств, каждое из которых представляет собой конвейер. Эффективная загрузка параллельно функционирующих конвейеров обеспечивается либо аппаратурой процессора, либо компилятором, на вход которого поступают программы на традиционном последовательном языке программирования, либо совместно аппаратурой и компилятором.

В компиляторах используется изощренная техника извлечения параллелизма из последовательных программ. Аппаратура микропроцессоров ориентирована на выделение более простых форм параллелизма, в том числе естественного. Стремление использовать присущий большинству программ естественный

параллелизм вычисления целочисленных адресных выражений и собственно обработки данных в формате с плавающей точкой привело к появлению разнесенных архитектур (decoupled architecture).

В первом приближении, микропроцессор с разнесенной архитектурой, как показано на рис.10, состоит из двух связанных подпроцессоров, каждый из которых управляется собственным потоком команд.

Условно эти подпроцессоры называются адресным А-процессором и исполнительным Е-процессором. А- и Е-процессоры имеют собственные наборы регистров А0,А1,... и Х0,Х1,..., соответственно и наборы команд. А-процессор выполняет все адресные вычисления и формирует обращения к памяти по чтению и записи. А-процессор является обыкновенным целочисленным процессором, поэтому он способен выполнять произвольные целочисленные преобразования, не связанные с вычислением адресов. Е-процессор реализует вычисления с плавающей точкой.

Данные, извлекаемые из памяти, используются либо в А-процессоре, будучи помещенными в FIFO очередь АА, либо помещаются в FIFO очередь, называемую АЕ очередью, для отсылки в Е-процессор. Когда Е-процессору требуются данные из памяти, он берет их из очереди АЕ. Если очередь пуста, то Е-процессор задерживается до поступления данных, что решает вопросы синхронизации работы А и Е-процессоров. Если Е-процессор выработал данное, которое должно быть отправлено в память, то он помещает его в FIFO очередь ЕА.

При записи данных в память после вычисления адреса А-процессор сразу отправляет адрес в FIFO очередь АW адресов записи в память, не дожидаясь, пока данные поступят в очередь ЕА. А-процессор группирует пары, выбирая первые элементы очередей ЕА и АW и отправляя эти пары в память. Естественно, если одна из очередей или обе пусты, то отсылка в память приостанавливается.

При чтении данных А-процессор отправляет адреса в память с указанием очередей АА или АЕ, в которые должны быть считаны данные из памяти.

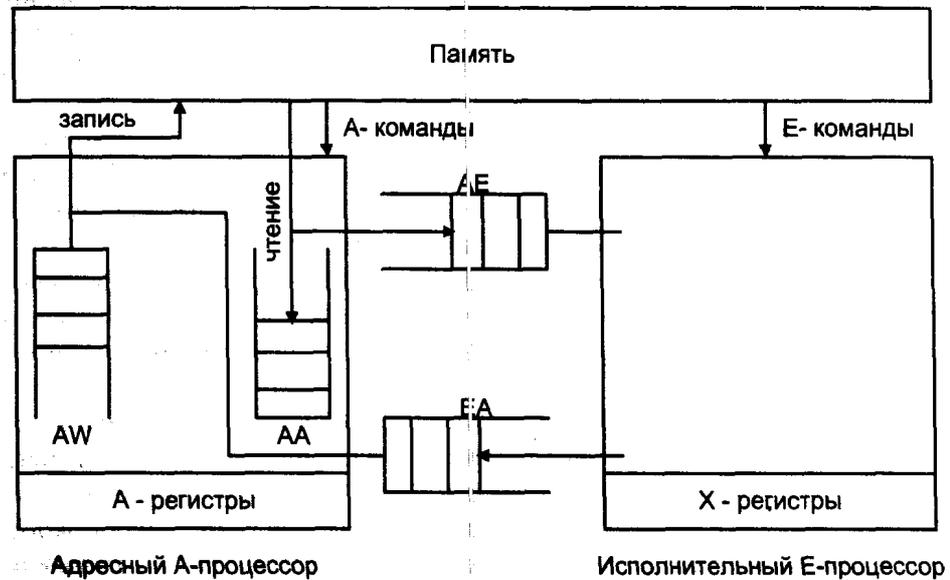


РИС.10. МИКРОПРОЦЕССОР С РАЗНЕСЕННОЙ АРХИТЕКТУРОЙ

Разнесенная архитектура позволяет достигать при скалярной обработке производительности, характерной для векторных процессоров, за счет предвыборки данных из памяти и автоматической развертки нескольких последовательных витков цикла в А-процессоре. Проблемы расщепления программы на программы для А- и Е-процессоров решаются на уровне компилятора или специальным блоком-расщепителем.

Важным системным аспектом разнесенной архитектуры служит интерфейс между процессором и памятью посредством транзакций чтения и записи. Это позволяет расположить между процессором и памятью произвольную коммутационную среду, что подводит логический базис под концепцию построения многопроцессорных систем.

Лекция 4.

Структурные методы уменьшения времени доступа к памяти

Совершенствование архитектуры микропроцессоров и механизмов доступа к памяти не может происходить без принятия совокупности соглашений о структуре программ и обрабатываемых данных. Это обусловлено, например, тем, что время доступа в основную память десять и более раз больше, чем время выполнения преобразований данных в регистрах процессора. Потому необходимо находить решения по уменьшению времени доступа, учитывающие этот факт.

Иерархическая структура памяти. Идеальная память должна обеспечивать процессор командами и данными так, чтобы не вызывать простоев процессора. При этом память должна иметь большую емкость. В современных условиях уменьшение времени доступа достигается введением многоуровневой иерархии памяти. Время доступа зависит от объема и типа используемой памяти.

Типовая современная иерархия памяти имеет следующую структуру:

- регистры 64 - 256 слов с временем доступа 1 такт процессора;
- кэш 1 уровня - 8к слов с временем доступа 1-2 такта;
- кэш 2 уровня - 256к слов с временем доступа 3-5 тактов;
- основная память - до 4 Гигаслов с временем доступа 12-55 тактов.

Используя помимо основной памяти небольшую и более быструю буферную память, можно значительно сократить количество обращений к основной памяти, за счет аккумуляции текущего фрагмента программного кода в буферной памяти. Создание иерархической многоуровневой памяти, пересылающей блоки программ и данных между уровнями памяти за время, пока предшествующие блоки обрабатываются процессором, позволяет существенно сократить простой процессора в ожидании данных. При этом эффект уменьшения времени доступа в память будет тем больше, чем больше время обработки данных в буферной памяти по сравнению с временем пересылки между буферной и основной памятью. Это достигается при локальности обрабатываемых данных, когда процессор многократно использует одни и те же данные для выработки некоторого результата. Например, такая ситуация имеет место при решении систем уравнений в научных и инженерных расчетах, когда короткие участки программного кода с большим количеством вложенных и зацепленных друг с другом циклов обрабатывают поочередно, переходя от точки к точке, небольшие порции данных, многократно используя одни и те же данные и внутренние результаты.

В связи с тем, что локально обрабатываемые данные могут возникать в динамике вычислений и не обязательно сконцентрированы в одной области при статическом размещении в основной памяти, буферную память организуют как ассоциативную, в которой данные содержатся в совокупности с их адресом в основной памяти. Такая буферная память получила название кэш-памяти. Кэш-память позволяет гибко согласовывать структуры данных, требуемые в динамике вычислений, со статическими структурами данных основной памяти.

Кэш имеет совокупность строк (cache-lines), каждая из которых состоит из фиксированного количества адресуемых единиц памяти (байтов, слов) с последовательными адресами. Типичный размер строки: 16, 64, 128, 256 байтов.

Наиболее часто используются три способа организации кэш-памяти, отличающиеся объемом аппаратуры, требуемой для их реализации. Это так называемые кэш-память с прямым отображением (direct-mapped cache), частично ассоциативная кэш-память (set-associative cache) и ассоциативная кэш-память (fully associative cache).

При использовании кэш-памяти с прямым отображением адрес представляется как набор трех компонент, составляющих группы старших, средних и младших разрядов адреса, соответственно тега, номера строки, смещения. Например, при 16-разрядном адресе старшие 5 разрядов могут представлять тег, следующие 7 разрядов - номер строки и последние 4 разряда - смещение в строке. В этом случае строка состоит из 6 адресуемых единиц памяти, всего

строк в кэше 128. Кэш-память с прямым отображением представляет собой набор строк, каждая из которых содержит компоненту тег и элементы памяти строки, адрес которых идентифицируется смещением относительно начала строки.

При этом устанавливается однозначное соответствие между адресом элемента памяти и возможным расположением этого элемента памяти в кэше, а именно: элемент памяти всегда располагается в строке, задаваемой компонентой "номер строки" адреса, и находится на позиции строки, задаваемой компонентой "смещение" адреса,

Наличие элемента данных по запрашиваемому адресу в кэше определяется значением тега. Если тег строки кэш-памяти равен компоненте "тег" адреса, то элемент данных содержится в кэш-памяти.

Иначе необходима подкачка в кэш-память строки, с заданным в адресе тегом.

Так как для определения наличия нужной строки данных в кэш-памяти требуется только одно сравнение тегов заданной строки и адреса, а само замещение строк выполняется по фиксированному местоположению, то объем оборудования, необходимый для реализации этого типа кэш-памяти, достаточно мал.

Недостатки этой организации - очевидны. Если программа использует поочередно элементы памяти из одной строки, но с различными значениями тегов, то это вызывает при каждом обращении замену строки с обращением к данным основной памяти.

Ассоциативная кэш-память использует двухкомпонентное представление адреса: группа старших разрядов трактуется как тег, а группа младших разрядов - как смещение в строке.

Нахождение строки в кэше определяется совпадением тега-строки со значением тега адреса. Количество строк в кэше может быть произвольным (естественное ограничение - количество возможных значений тегов). Поэтому при определении нахождения требуемой строки в кэш-памяти необходимо сравнение тега адреса с тегами всех строк кэша. Если выполнять это последовательно, строка за строкой, то время выполнения сравнений будет непозволительно большим. Поэтому сравнение выполняется параллельно во всех строках с использованием принципов построения ассоциативной памяти, что и дало название этому способу организации кэш-памяти.

При отсутствии необходимой строки в кэш-памяти одна из его строк должна быть заменена на требуемую. Используются разнообразные алгоритмы определения заменяемой строки, например циклический, замена наиболее редко используемой строки, замена строки, к которой дольше всего не было обращений, и другие.

Частично-ассоциативная кэш-память комбинирует оба вышеописанных подхода: кэш-память состоит из набора ассоциативных блоков кэш-памяти. Средняя компонента адреса задает в отличие от прямо адресуемой кэш-памяти

не номер строки, а номер одного из ассоциативных блоков. При поиске данных ассоциативное сравнение тегов выполняется только для набора блоков (возможна организация кэша, когда таких наборов несколько), номер которого совпадает со средней компонентой адреса. По количеству n строк в наборе кэш-память называется n -входовой.

Соответствие между данными в оперативной памяти и кэш-памяти обеспечивается внесением изменений в те области оперативной памяти, для которых данные в кэш-памяти подверглись модификации. Соответствие данных обеспечивается параллельно с основными вычислениями. Существует несколько способов его реализации (и, соответственно, несколько режимов работы кэш-памяти).

Один способ предполагает внесение изменений в оперативную память сразу после изменения данных в кэше. При этом процессор простаивает в ожидании завершения записи в основную память. В основной памяти поддерживается правильная копия данных кэша, и при замене строк не требуется никаких дополнительных действий. Кэш-память, работающая в таком режиме, называется памятью со сквозной записью (write-through).

Другой способ предполагает отображение изменений в основной памяти только в момент вытеснения строки данных из кэша. Если данные по адресу памяти, в который необходимо произвести запись, находятся в кэш-памяти, то идет запись только в кэш-память. При отсутствии данных в кэш-памяти производится запись в основную память. Такой режим работы кэша получил название обратной записи (write-back).

Существуют также промежуточные варианты (buffered write through), при которых запросы на изменение в основной памяти буферизуются и не задерживают процессор на время операции записи в память. Эта запись выполняется по мере возможности доступа контроллера кэш-памяти к основной памяти.

В процессорах, предусматривающих возможность использования в мультипроцессорных конфигурациях, применяется более сложный протокол MESI (Modified, Exclusive, Shared, Invalid) организации кэш-памяти с обратной записью, который предотвращает лишние передачи данных между кэш-памятью и основной памятью.

Для мультипроцессорных систем, в которых память физически распределена между процессорными модулями, идентичность данных в кэшах (когерентность кэшей) различных модулей поддерживается с помощью межмодульных пересылок. Существует несколько основных подходов.

Прямолинейный подход к поддержанию когерентности кэшей в мультипроцессорной системе заключается в том, что при каждом непопадании в кэш в любом процессоре инициируется запрос требуемой строки из того блока памяти, в котором эта строка размещена. В дальнейшем этот блок памяти будет по отношению к этой строке называться резидентным. Запрос передается через коммутатор в модуль с резидентным для строки блоком памяти, из которого

затем необходимая строка через коммутатор присылается в модуль, в котором произошло непопадание. Таким образом, в частности, обеспечивается начальное заполнение кэшей. При этом в каждом модуле для каждой резидентной строки ведется список модулей, в кэшах которых эта строка размещается. Строка, размещенная в кэше более чем одного модуля, в дальнейшем будет называться разделяемой.

Собственно когерентность кэшей обеспечивается следующим. При обращении к кэш-памяти в ходе операции записи данных, после самой записи, процессор приостанавливается до тех пор, пока не выполнится последовательность действий: измененная строка кэша пересылается в резидентную память модуля, затем, если строка была разделяемой, она пересылается из резидентной памяти во все модули, указанные в списке модулей, разделяющих эту строку. После получения подтверждений, что все копии изменены, резидентный модуль пересылает в процессор, приостановленный после записи, разрешение продолжать вычисления.

Изложенный алгоритм обеспечения когерентности хотя и является логически работоспособным, однако практически редко применяется из-за больших простоев процессоров при операциях записи в кэш строки. На практике применяют более сложные алгоритмы, обеспечивающие меньшие простои процессоров.

Для изложения одного из алгоритмов поддержки когерентности кэшей, известного как DASH, зададим некоторые начальные условия и введем определения.

Каждый модуль имеет для каждой строки, резидентной в модуле, список модулей, в кэшах которых размещены копии строк.

С каждой строкой в резидентном для нее модуле свяжем три ее возможных глобальных состояния:

- 1) "некэшированная", если копия строки не находится в кэше какого-либо другого модуля, кроме, возможно, резидентного для этой строки;
- 2) "удаленно-разделенная", если копии строки размещены в кэшах других модулей;
- 3) "удаленно-измененная", если строка изменена операцией записи в каком-либо модуле.

Кроме этого, каждая строка кэша может находиться в одном из трех локальных состояний:

- 1) "невозможная к использованию";
- 2) "разделяемая", если есть неизменная копия, которая, возможно, размещается также в других кэшах;
- 3) "измененная", если копия изменена операцией записи,

Перейдем к изложению алгоритма. Каждый процессор может читать из своего кэша, если состояние читаемой строки "разделяемая" или "измененная".

Если строка отсутствует в кэше или находится в состоянии "невозможная к использованию", то посылается запрос "промах чтения", который направляется в модуль, резидентный для требуемой строки.

Бели глобальное состояние строки в резидентном модуле - "некэшированная" или "удаленно-разделенная", то копия строки посылается в запросивший модуль, а в список модулей, содержащих копии рассматриваемой строки, вносится модуль, запросивший копию.

Если состояние строки "удаленно-измененная", то запрос "промах чтения" перенаправляется в модуль, содержащий измененную строку. Этот модуль пересылает требуемую строку в запросивший модуль и в модуль, резидентный для этой строки, и устанавливает в резидентном модуле для этой строки состояние удаленно-распределенная .

Если процессор выполняет операцию записи и состояние строки, в которую производится запись, - "измененная", то запись выполняется и вычисления продолжают. Если состояние строки - "невозможная к использованию" или "разделяемая", то модуль посылает в резидентный для строки модуль запрос на захват в исключительное использование этой строки и приостанавливает выполнение записи до получения подтверждений, что все остальные модули, разделяющие с ним рассматриваемую строку, перевели ее копии в состояние "невозможная к использованию".

Если глобальное состояние строки в резидентном модуле - "некэшированная", то строка отсылается запросившему модулю, и этот модуль продолжает приостановленные вычисления .

Если глобальное состояние строки - "удаленно-разделенная", то резидентный модуль рассылает по списку всем модулям, имеющим копию строки, запрос на переход этих строк в состояние "невозможная к использованию". По получении этого запроса каждый из модулей изменяет состояние своей копии строки на "невозможная к использованию" и посылает подтверждение исполнения в модуль, инициировавший операцию записи. При этом в приостановленном модуле строка после исполнения записи переходит в состояние "удаленно-измененная".

Предпринимаются попытки повысить эффективность реализации алгоритма когерентности, в частности, за счет учета специфики параллельных программ, в которых используются асинхронно одни и те же данные на каждом временном интервале исключительно одним процессором с последующим переходом обработки к другому процессору. Такого рода ситуации случаются, например, при определении условий окончания итераций, В этом случае возможна более эффективная схема передачи строки из кэша одного процессора в кэш другого процессора.

В системах, использующих коммутатор с временным разделением (шину), интерфейс с шиной каждого модуля "прослушивает" все передачи по шине, поэтому нет необходимости вести списки модулей, разделяющих строки.

Кэш-память с обратной записью создает меньшую нагрузку на шину

процессора и обеспечивает большую производительность, однако контроллер для write-back кэша значительно сложнее.

Контроллер кэша отслеживает адреса памяти, выдаваемые процессором, и если адрес соответствует данным, содержащимся в одной из строк кэша, то отмечается "попадание в кэш", и данные из кэша направляются в процессор. Если данных в кэше не оказывается, то фиксируется "промах", и инициируются действия по доставке в кэш из памяти требуемой строки. В ряде процессоров, выполняющих одновременно совокупность команд, допускается несколько промахов, прежде чем будет запущен механизм замены строк.

Рассуждения о том, какой способ организации кэш-памяти более предпочтителен, должны учитывать особенности генерации программ компилятором, а также использование программистом при подготовке программы сведений о работе компилятора и контроллера кэш-памяти. То есть более простой способ организации кэш-памяти, поддерживаемый компилятором, при исполнении программ, написанных в соответствии с некоторыми правилами, обусловленными особенностями компиляции и организации кэш-памяти, может дать лучший результат, чем сложный способ организации кэш-памяти.

Так как области памяти программ и данных различны и к ним происходит одновременный доступ, то для повышения параллелизма при работе с памятью делают отдельные кэши команд и данных.

Расслоение памяти. Другим структурным способом уменьшения времени доступа к памяти служит расслоение памяти. В предположении, что выборка из памяти выполняется по последовательным адресам возможно использование k блоков памяти с размещением в блоке i , где $i=0, \dots, k-1$, слов с адресами $n=i+krrp \bmod k$, где $p=0,1, \dots, M$. В этом случае возможно k параллельных обращений в память по адресам, принадлежащим различным блокам. Поэтому выборка команд программы за исключением команд, выбираемых как результат ветвления, может быть ускорена применением расслоения памяти. Аналогично может быть ускорена обработка массивов данных, последовательные элементы которых помещаются в разные блоки памяти.

Многоуровневая иерархия и расслоение памяти могут использоваться совместно.

CISC- и RISC- процессоры.

АППАРАТНЫЕ ВОЗМОЖНОСТИ ПРОЦЕССОРОВ ОПРЕДЕЛЯЮТСЯ УРОВНЕМ РАЗВИТИЯ МИКРОЭЛЕКТРОННОЙ ТЕХНОЛОГИИ, КОТОРАЯ ОГРАНИЧИВАЕТ ЧИСЛО ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ, РАЗМЕЩАЕМЫХ НА КРИСТАЛЛЕ. ФАКТИЧЕСКИ ЕГО РАЗМЕРЫ И ЧИСЛО РАЗМЕЩАЕМЫХ НА НЕМ ТРАНЗИСТОРОВ ПОЧТИ ЦЕЛИКОМ ОПРЕДЕЛЯЮТ ТОТ НАБОР АППАРАТНЫХ УСТРОЙСТВ, КОТОРЫЕ ВКЛЮЧАЮТСЯ В СОЗДАВАЕМЫЙ ПРОЦЕССОР. РАЗРАБОТЧИКИ СТРЕМЯТСЯ ВВЕСТИ В ЕГО СОСТАВ КАК МОЖНО БОЛЬШЕЕ ЧИСЛО РАЗЛИЧНЫХ АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ УСТРОЙСТВ, СОПРОЦЕССОРОВ И Т.Д., ПОСКОЛЬКУ ОЧЕВИДНО, ЧТО АППАРАТНАЯ РЕАЛИЗАЦИЯ КАКОЙ-ЛИБО ФУНКЦИИ ОБЕСПЕЧИВАЕТ БОЛЕЕ БЫСТРОЕ ЕЕ ВЫПОЛНЕНИЕ, ЧЕМ ПРОГРАММНАЯ. ДА И ДЛИНА КОДА ДЛЯ РЕАЛИЗУЕМЫХ АППАРАТНЫМ ПУТЕМ ФУНКЦИЙ СУЩЕСТВЕННО МЕНЬШЕ, ПОЭТОМУ В 80-Х ГОДАХ ПРОЦЕССОРЫ "РОСЛИ ВШИРЬ", ОСНАЩАЛИСЬ НОВЫМИ УЗЛАМИ, СОВЕРШЕНСТВУЯ УЖЕ ИМЕЮЩИЕСЯ. КРОМЕ ТОГО, РАЗРАБОТЧИКАМ

ПРОЦЕССОРОВ ПРИХОДИЛОСЬ ПОСТОЯННО ПОДСТРАИВАТЬСЯ ПОД СТРЕМЛЕНИЕ ПРОГРАММИСТОВ ИМЕТЬ ВОЗМОЖНО БОЛЕЕ МОЩНУЮ И ГИБКУЮ СИСТЕМУ КОМАНД. ТАКИЕ СИСТЕМЫ КОМАНД С РАЗНООБРАЗНЫМИ СПОСОБАМИ АДРЕСАЦИИ И НАБОРАМИ УСЛОВНЫХ ПЕРЕХОДОВ И ВЫЗОВОВ ПОДПРОГРАММ, СТРОКОВЫМИ ОПЕРАЦИЯМИ И ПРЕФИКСАЦИЯМИ ОБЕСПЕЧИВАЛИ СОЗДАНИЕ КОРОТКИХ ПРОГРАММ, ГАРАНТИРОВАЛИ МЕНЬШЕЕ ЧИСЛО ОШИБОК В НИХ. И РАЗРАБОТЧИКИ НЕ МОГЛИ НЕ СЧИТАТЬСЯ С ЭТИМ.

! ПОД АРХИТЕКТУРОЙ В ДАННОМ СЛУЧАЕ СЛЕДУЕТ ПОНИМАТЬ ТОЛЬКО СИСТЕМУ КОМАНД, ИГРАЮЩУЮ РОЛЬ ИНТЕРФЕЙСА МЕЖДУ АППАРАТНЫМ И ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ.

ПЕРВЫЕ РОСТКИ ПРОТИВОРЕЧИЙ, ПРИВЕДШИХ К ТОМУ, ЧТО ПУТИ ЗАНИМАЮЩИХСЯ РАЗРАБОТКОЙ И ПРОИЗВОДСТВОМ МП КОМПАНИЙ РАЗОШЛИСЬ В РАЗНЫЕ СТОРОНЫ, ОБНАРУЖИЛИСЬ ДАВНО. ЗА ПЕРВЫЕ 10 ЛЕТ СУЩЕСТВОВАНИЯ МП СФОРМИРОВАЛАСЬ АРХИТЕКТУРА CISC (COMPLEX INSTRUCTION SET COMPUTING — ВЫЧИСЛЕНИЯ СО СЛОЖНЫМ НАБОРОМ ИНСТРУКЦИЙ). В ПОСТРОЕННЫХ ПО ЕЁ ПРИНЦИПАМ ПРОЦЕССОРАХ БОЛЕЕ 2/3 КРИСТАЛЛА ЗАНИМАЮТ ЛОГИКА, РЕАЛИЗУЮЩАЯ ВЫБОРКУ И ДЕКОДИРОВАНИЕ СЛОЖНЫХ ИНСТРУКЦИЙ, ПРЯМОЕ ВЗАИМОДЕЙСТВИЕ НЕ ТОЛЬКО С РЕГИСТРОВОЙ ПАМЯТЬЮ, НО И С ВНЕШНЕЙ, А ТАКЖЕ ИЗОЩРЁННЫЕ СХЕМЫ АДРЕСАЦИИ. ПРИ ЭТОМ БЛОКИ, НЕПОСРЕДСТВЕННО ЗАНЯТЫЕ ВЫЧИСЛЕНИЯМИ ЗАНИМАЮТ ЛИШЬ 1/3 ПЛОЩАДИ КРИСТАЛЛА. НА ПРОТЯЖЕНИИ НЕСКОЛЬКИХ ДЕСЯТИЛЕТИЙ СИСТЕМЫ КОМАНД ЭВОЛЮЦИОНИРОВАЛИ, НАХОДЯСЬ В СЛОЖНОЙ ЗАВИСИМОСТИ ОТ СТОИМОСТИ АППАРАТНЫХ РЕСУРСОВ, И ПРЕЖДЕ ВСЕГО САМОГО ДОРОГОГО ИЗ НИХ – ОПЕРАТИВНОЙ ПАМЯТИ. ИНЖЕНЕРАМИ РУКОВОДИЛО СТРЕМЛЕНИЕ СОКРАТИТЬ РАЗМЕР ПРОГРАММ, ДЛЯ ЭТОГО ОНИ СТАРАЛИСЬ ВЛОЖИТЬ КАК МОЖНО БОЛЬШЕ ФУНКЦИОНАЛЬНОСТИ В ОДНУ КОМАНДУ. К НАЧАЛУ 80-Х ЧИСЛО ТАКИХ КОМАНД ДОСТИГЛО ПОЧТИ 300 СОТЕН, И ПОЧТИ КАЖДАЯ ИЗ НИХ ИМЕЛА ДО 6 ТИПОВ АДРЕСАЦИИ. К ТОМУ ЖЕ КОМАНДЫ НЕ ИМЕЛИ ФИКСИРОВАННОЙ ДЛИНЫ.

ОДНАКО, НАЧИНАЯ С НЕКОТОРОГО МОМЕНТА CISC-ИДЕОЛОГИЯ СТАЛА НЕ УСКОРИТЕЛЕМ, А ТОРМОЗОМ РОСТА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ. ДЕЛО В ТОМ, ЧТО БОЛЬШОЕ РАЗНООБРАЗИЕ КОМАНД И РАЗЛИЧНАЯ ИХ ДЛИНА СУЩЕСТВЕННО УСЛОЖНЯЮТ АППАРАТУРУ УПРАВЛЕНИЯ ВНУТРИ ПРОЦЕССОРА. ТАК, НАПРИМЕР, ОНА ДОЛЖНА ОПРЕДЕЛИТЬ ДЛИНУ КОМАНДЫ (МОЖЕТ ВАРЬИРОВАТЬСЯ В ПРЕДЕЛАХ ОТ ОДНОГО ДО ПЯТИ-ШЕСТИ БАЙТ), ВЫДЕЛИТЬ КОД ОПЕРАЦИИ, ПОДГОТОВИТЬ ОПЕРАНДЫ, КОТОРЫЕ МОГУТ НАХОДИТЬСЯ КАК В РЕГИСТРАХ, ТАК И В ОПЕРАТИВНОЙ ПАМЯТИ. РЕЗУЛЬТАТ ВЫЧИСЛЕНИЙ ТАКЖЕ НУЖНО ПЕРЕНЕСТИ ИЛИ В ПАМЯТЬ, ИЛИ В КАКОЙ-ЛИБО ИЗ РЕГИСТРОВ.

ОСОЗНАВ УКАЗАННЫЕ ФАКТЫ, МНОГИЕ ИССЛЕДОВАТЕЛИ УЖЕ В 80-Е ГОДЫ НАЧАЛИ ПОИСКИ АЛЬТЕРНАТИВНЫХ ПУТЕЙ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ПРОЦЕССОРОВ. БЫЛО ЯСНО, ЧТО НУЖНО ПОПЫТАТЬСЯ УМЕНЬШИТЬ «НАКЛАДНЫЕ РАСХОДЫ», ОТЪЕДАЮЩИЕ ДО ДВУХ ТРЕТЕЙ ПОЛЕЗНОЙ ПЛОЩАДИ КРИСТАЛЛА, СНИЗИВ ЧИСЛО ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ В НИХ ДО МИНИМУМА. ДЛЯ ЭТОГО БЫЛО НЕОБХОДИМО ПЕРЕНЕСТИ "ЦЕНТР ТЯЖЕСТИ" НА ПРОГРАММНЫЕ СРЕДСТВА, ОСТАВИВ МИНИМАЛЬНО ВОЗМОЖНОЕ ЧИСЛО КОМАНД, ИМЕЮЩИХ ПРОСТУЮ И РЕГУЛЯРНУЮ СТРУКТУРУ. ДРУГИМИ СЛОВАМИ, НАЧАЛА ФОРМИРОВАТЬСЯ СТРАТЕГИЯ "ДЛИННАЯ ПРОГРАММА — КОРОТКИЕ КОМАНДЫ" В ПРОТИВОВЕС ГОСПОДСТВУЮЩЕЙ «КОРОТКАЯ ПРОГРАММА — ДЛИННЫЕ КОМАНДЫ». ОСТАВАЛОСЬ ТОЛЬКО НАЙТИ РАЗУМНЫЙ КОМПРОМИСС МЕЖДУ РЕСУРСАМИ, ОТВОДИМЫМИ ОСНОВНЫМ ВЫЧИСЛИТЕЛЬНЫМ УЗЛАМ ПРОЦЕССОРА И УСТРОЙСТВАМ УПРАВЛЕНИЯ.

И ВСКОРЕ ЭТОТ КОМПРОМИСС БЫЛ НАЙДЕН. ВНАЧАЛЕ ДЖОН КУК ИЗ IBM RESEARCH LABS НА СВОЕМ ЭКСПЕРИМЕНТАЛЬНОМ КОМПЬЮТЕРЕ "MODEL 801" ПОКАЗАЛ, ЧТО ПРИ ИСПОЛЬЗОВАНИИ В ПРОГРАММЕ ЛИШЬ ПРОСТЕЙШИХ КОМАНД ФОРМАТА «РЕГИСТР—

РЕГИСТР» (ОПЕРАНДЫ ИЗВЛЕКАЮТСЯ ИСКЛЮЧИТЕЛЬНО ИЗ РЕГИСТРОВ И ПОСЛЕ ВЫПОЛНЕНИЯ ОПЕРАЦИИ ПОМЕЩАЮТСЯ ТОЛЬКО В РЕГИСТРЫ) СКОРОСТЬ ВЫПОЛНЕНИЯ БОЛЬШИНСТВА ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ ВОЗРАСТАЕТ В ДВА-ТРИ РАЗА. ЗАТЕМ УЧЕНЫМИ СТЕНФОРДСКОГО УНИВЕРСИТЕТА БЫЛО ПОКАЗАНО, ЧТО ИСКЛЮЧЕНИЕ ИЗ НАБОРА КОМАНД СЛОЖНЫХ ОПЕРАЦИЙ УМЕНЬШАЕТ ЧИСЛО ЭЛЕМЕНТОВ ПРОЦЕССОРА НА ПОРЯДОК ЗА СЧЕТ УПРОЩЕНИЯ СХЕМ УПРАВЛЕНИЯ РЕСУРСАМИ ПРОЦЕССОРА (ПРИ ЭТОМ ПРОИЗВОДИТЕЛЬНОСТЬ СИСТЕМЫ ОСТАЕТСЯ ПРАКТИЧЕСКИ НЕИЗМЕННОЙ).

ОПЫТ МНОГОЧИСЛЕННЫХ ИССЛЕДОВАТЕЛЕЙ СТАТИСТИЧЕСКИХ СВОЙСТВ ПРОГРАММ ВЫЯВИЛ, ЧТО 80 % ПРОЦЕССОРНОГО ВРЕМЕНИ ОБЫЧНО ТРАТИТСЯ НА ВЫПОЛНЕНИЕ ВСЕГО 20 % ОТ ОБЩЕГО ЧИСЛА ИНСТРУКЦИЙ ОБЫЧНЫХ CISC-ПРОЦЕССОРОВ. ПРИЧЕМ ВСЕ ЭТИ 20 % ПРИХОДЯТСЯ НА КОМАНДЫ ТИПА "РЕГИСТР—РЕГИСТР".

И НАКОНЕЦ, В РАБОТАХ Д. ПАТТЕРСОНА И К. СЕКУИНА БЫЛИ СФОРМУЛИРОВАНЫ ЧЕТЫРЕ ОСНОВНЫХ ПРИНЦИПА, КОТОРЫЕ ДОЛЖНЫ БЫТЬ ПОЛОЖЕНЫ В ОСНОВУ ПРОЦЕССОРОВ ГРУППЫ RISC (REDUCED INSTRUCTION SET COMPUTING – ВЫЧИСЛЕНИЯ С СОКРАЩЕННЫМ НАБОРОМ КОМАНД):

А) ЛЮБАЯ ОПЕРАЦИЯ, К КАКОМУ БЫ ТИПУ ОНА НИ ПРИНАДЛЕЖАЛА, ДОЛЖНА ВЫПОЛНЯТЬСЯ ЗА ОДИН ТАКТ;

Б) ОПЕРАЦИИ ОБРАБОТКИ ДАННЫХ РЕАЛИЗУЮТСЯ ТОЛЬКО В ФОРМАТЕ “РЕГИСТР—РЕГИСТР”. ОБМЕН МЕЖДУ ПАМЯТЬЮ И РЕГИСТРАМИ (Т. Е. МОДИФИКАЦИЯ ПАМЯТИ И ЧТЕНИЕ ИЗ НЕЕ НЕОБХОДИМЫХ ДАННЫХ) ОСУЩЕСТВЛЯЕТСЯ ТОЛЬКО ЛИШЬ С ПОМОЩЬЮ СПЕЦИАЛЬНЫХ КОМАНД ЧТЕНИЯ/ЗАПИСИ;

В) СИСТЕМА КОМАНД ДОЛЖНА СОДЕРЖАТЬ МИНИМАЛЬНОЕ ЧИСЛО НАИБОЛЕЕ ЧАСТО ИСПОЛЪЗУЕМЫХ ПРОСТЕЙШИХ КОМАНД ОДИНАКОВОЙ ДЛИНЫ:

Г) СОСТАВ СИСТЕМЫ КОМАНД ДОЛЖЕН БЫТЬ ОПТИМИЗИРОВАН С УЧЕТОМ ТРЕБОВАНИЙ КОМПИЛЯТОРОВ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ.

КСТАТИ, НАД ИДЕЯМИ, ВОШЕДШИМИ В КОНЦЕПЦИЮ RISC, ТРУДИЛИСЬ НЕ ТОЛЬКО НА ЗАПАДЕ. ЕЩЁ В 70-Х ГОДАХ, КОГДА АМЕРИКАНСКИЕ УЧЁНЫЕ ТОЛЬКО ПРИБЛИЗИЛИСЬ К ФОРМУЛИРОВКЕ ТОГО, ЧТО СТАЛО НАЗЫВАТЬСЯ RISC, В СОВЕТСКОМ СОЮЗЕ УЖЕ БЫЛИ ПОСТРОЕНЫ БОЛЬШИЕ ЭВМ, В КОТОРЫХ БЫЛИ РЕАЛИЗОВАНЫ РЕШЕНИЯ, ПОЗВОЛИВШИЕ ИСПОЛЬЗОВАТЬ RISC-АРХИТЕКТУРУ В ОБЛАСТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ. НАПРИМЕР, СУПЕРСКАЛЯРНАЯ АРХИТЕКТУРА БЫЛА РЕАЛИЗОВАНА У НАС 1978 ГОДУ, В ТО ВРЕМЯ КАК АМЕРИКАНЦЫ ПОДОШЛИ К НЕЙ ЛИШЬ В НАЧАЛЕ 90-Х ГОДОВ. А ВЕДЬ ИМЕННО СУПЕРСКАЛЯРНОСТЬ НАЗЫВАЮТ СЕГОДНЯ 5-ЫМ КРИТЕРИЕМ ПРИНАДЛЕЖНОСТИ ПРОЦЕССОРА К RISC-АРХИТЕКТУРЕ.

НАИБОЛЕЕ СЛОЖНО ВЫПОЛНИМОЕ ИЗ УПОМЯНУТЫХ ТРЕБОВАНИЙ — ПЕРВОЕ, В СООТВЕТСТВИИ С КОТОРЫМ, КАЖДАЯ КОМАНДА ДОЛЖНА ВЫПОЛНЯТЬСЯ НЕ БОЛЕЕ ЧЕМ ЗА ОДИН ТАКТ. ОДНАКО, НЕСМОТРИ НА СУЩЕСТВЕННОЕ УПРОЩЕНИЕ, НЕОБХОДИМО, ЧТОБЫ УСТРОЙСТВО УПРАВЛЕНИЯ ПО-ПРЕЖНЕМУ ВЫПОЛНЯЛО ОПЕРАЦИИ ПО ЧТЕНИЮ КОДА КОМАНДЫ, ДЕКОДИРОВАНИЮ, ПОДГОТОВКЕ ОПЕРАНДОВ, СОБСТВЕННО ВЫПОЛНЕНИЮ КОМАНДЫ И ПЕРЕМЕЩЕНИЮ РЕЗУЛЬТАТА В СООТВЕТСТВУЮЩИЙ РЕГИСТР. ЕСЛИ УЧЕСТЬ, ЧТО КАЖДАЯ ИЗ ЭТИХ ОПЕРАЦИЙ ОБЯЗАНА СИНХРОНИЗИРОВАТЬСЯ С ФРОНТОМ (ИЛИ СПАДОМ) СИГНАЛА ТАКТОВОГО ГЕНЕРАТОРА, ТО ЕДИНСТВЕННО ВОЗМОЖНЫЙ ВАРИАНТ РЕШЕНИЯ ПРОБЛЕМЫ - ИСПОЛЬЗОВАНИЕ ТАК НАЗЫВАЕМОГО КОНВЕЙЕРА КОМАНД. ПРИ ЭТОМ ПО ПЕРВОМУ ПЕРЕПАДУ СИГНАЛА ТАКТОВОГО ГЕНЕРАТОРА В СООТВЕТСТВУЮЩЕМ УСТРОЙСТВЕ ВЫДЕЛЯЕТСЯ КОД ОПЕРАЦИИ, КОТОРЫЙ ЗАТЕМ ПЕРЕДАЕТСЯ В УСТРОЙСТВО ДЕКОДИРОВАНИЯ; ПО ВТОРОМУ - В УСТРОЙСТВЕ ИЗВЛЕЧЕНИЯ ВЫДЕЛЯЕТСЯ КОД ОПЕРАЦИИ СЛЕДУЮЩЕЙ КОМАНДЫ, А В УСТРОЙСТВЕ ДЕКОДИРОВАНИЯ ПРОИСХОДИТ

ДЕКОДИРОВАНИЕ ПЕРВОЙ КОМАНДЫ И ПЕРЕДАЧА ЕЕ В СООТВЕТСТВУЮЩЕЕ ИСПОЛНИТЕЛЬНОЕ УСТРОЙСТВО - АЛУ, СОПРОЦЕССОР И Т. Д. ТРЕТИЙ ПЕРЕПАД СОПРОВОЖДАЕТСЯ ИЗВЛЕЧЕНИЕМ КОДА ОПЕРАЦИИ ТРЕТЬЕЙ КОМАНДЫ, ДЕКОДИРОВАНИЕМ ВТОРОЙ И ПОДГОТОВКОЙ ДАННЫХ ДЛЯ ВЫПОЛНЕНИЯ ПЕРВОЙ.

ТАКИМ ОБРАЗОМ, ПО КАЖДОМУ ИЗ ТАКОВЫХ ИМПУЛЬСОВ НА КОНВЕЙЕР ВСТУПАЕТ НОВАЯ КОМАНДА, И НЕСКОЛЬКО УЖЕ ОБРАБАТЫВАЮТСЯ НА РАЗНЫХ ЕГО СТУПЕНЯХ. ОДНОВРЕМЕННО, ТАКЖЕ ПО КАЖДОМУ ИЗ ТАКОВЫХ ИМПУЛЬСОВ, ЕГО ПОКИДАЕТ ОДНА ВЫПОЛНЕННАЯ КОМАНДА. И ХОТЯ НА ВЫПОЛНЕНИЕ КАЖДОЙ ЗАТРАЧИВАЕТСЯ ПО-ПРЕЖНЕМУ ОТ ЧЕТЫРЕХ ДО СЕМИ-ВОСЬМИ ТАКТОВ (В РАССМОТРЕННОМ ВЫШЕ УСЛОВНОМ КОНВЕЙЕРЕ - ПЯТЬ), КАЖДЫЙ ИЗ НИХ СОПРОВОЖДАЕТСЯ, КАК ЭТО И ТРЕБОВАЛОСЬ, ВЫПОЛНЕНИЕМ ОДНОЙ КОМАНДЫ. СЛЕДОВАТЕЛЬНО, ЕСЛИ ДЛЯ RISC-ПРОЦЕССОРОВ КОНВЕЙЕР КОМАНД ЯВЛЯЕТСЯ НЕОБЯЗАТЕЛЬНЫМ (ХОТЯ И ЖЕЛАТЕЛЬНЫМ) ЭЛЕМЕНТОМ, ТО ДЛЯ RISC-ПРОЦЕССОРОВ ОН ОБЯЗАТЕЛЕН. ОТМЕТИМ, ЧТО БОЛЬШИНСТВО RISC-ПРОЦЕССОРОВ ИМЕЮТ НЕ ОДИН, А НЕСКОЛЬКО (ОТ ДВУХ ДО ЧЕТЫРЕХ) КОНВЕЙЕРОВ, ЗА ЧТО ОНИ ПОЛУЧИЛИ НАЗВАНИЕ СУПЕРСКАЛЯРНЫХ (В ОТЛИЧИЕ ОТ СКАЛЯРНЫХ - ОДНОКОНВЕЙЕРНЫХ).

СЛЕДУЮЩАЯ ОСОБЕННОСТЬ RISC-ПРОЦЕССОРОВ - ВЫСОКАЯ СТЕПЕНЬ ДРОБЛЕНИЯ КОНВЕЙЕРА. В РАССМОТРЕННОМ ВЫШЕ ПРИМЕРЕ ОН СОСТОИТ ИЗ ПЯТИ СТУПЕНЕЙ: ИЗВЛЕЧЕНИЯ КОДА ОПЕРАЦИИ, ДЕКОДИРОВАНИЯ, ПОДГОТОВКИ ОПЕРАНДОВ, ИСПОЛНЕНИЯ, СОХРАНЕНИЯ РЕЗУЛЬТАТА. РЕАЛЬНО RISC-ПРОЦЕССОРЫ ХАРАКТЕРИЗУЮТСЯ СЕМИДЕСЯТИ СТУПЕНЧАТЫМИ КОНВЕЙЕРАМИ. С УВЕЛИЧЕНИЕМ ЧИСЛА СТУПЕНЕЙ ДЕЙСТВИЯ, ВЫПОЛНЯЕМЫЕ НА КАЖДОЙ ИЗ НИХ, ВСЕ БОЛЕЕ И БОЛЕЕ УПРОЩАЮТСЯ. ПОСЛЕДНЕЕ ОЗНАЧАЕТ, ЧТО УМЕНЬШАЕТСЯ ЧИСЛО НЕОБХОДИМЫХ ДЛЯ ЭТОГО ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ И ПОЯВЛЯЕТСЯ ВОЗМОЖНОСТЬ ПОВЫСИТЬ ТАКОВУЮ ЧАСТОТУ ПРОЦЕССОРА.

ТРЕТЬЯ ОСОБЕННОСТЬ, ВЫТЕКАЮЩАЯ ИЗ ПРЕДЫДУЩЕЙ, - РАЗВИТЫЕ СРЕДСТВА ПРОГНОЗИРОВАНИЯ ВЕТВЛЕНИЙ И ПЕРЕХОДОВ. В ПРОГРАММАХ ДЛЯ СЕМЕЙСТВА X86 КОМАНДА ПЕРЕХОДА ВСТРЕЧАЕТСЯ В СРЕДНЕМ ЧЕРЕЗ КАЖДЫЕ ШЕСТЬ, В ПРОГРАММАХ ДЛЯ RISC-ПРОЦЕССОРОВ, КОМАНДЫ КОТОРЫХ ПРОЩЕ, - ЧЕРЕЗ КАЖДЫЕ 10... 12 КОМАНД. ВСТРЕТИВ КОМАНДУ УСЛОВНОГО ПЕРЕХОДА, ПРОЦЕССОР ДОЛЖЕН СДЕЛАТЬ ПРЕДПОЛОЖЕНИЕ О ТОМ, ВЫПОЛНИТСЯ УСЛОВИЕ ИЛИ НЕТ, И В ЗАВИСИМОСТИ ОТ ЭТОГО НАЧАТЬ ПРЕДВЫБОРКУ КОМАНД ЛИБО С АДРЕСА ПРЕДПОЛАГАЕМОГО ПЕРЕХОДА, ЛИБО С АДРЕСА, СЛЕДУЮЩЕГО ЗА ТЕКУЩИМ. ЕСЛИ ПЕРЕХОД ПРЕДСКАЗАН НЕПРАВИЛЬНО, ПРОЦЕССОРУ НЕОБХОДИМО УДАЛИТЬ СО ВСЕХ СТУПЕНЕЙ КОНВЕЙЕРА КОМАНДЫ, ОТНОСЯЩИЕСЯ К НЕВЕРНО СДЕЛАННОМУ ПРЕДПОЛОЖЕНИЮ, И ПЕРЕЗАГРУЗИТЬ КОНВЕЙЕР. ЭТО ОСОБЕННО СКАЗЫВАЕТСЯ НА РАБОТЕ СУПЕРСКАЛЯРНЫХ ПРОЦЕССОРОВ - НА РАЗНЫХ СТУПЕНЯХ ТРЕХ-ЧЕТЫРЕХ КОНВЕЙЕРОВ МОЖЕТ НАХОДИТЬСЯ ДОВОЛЬНО МНОГО КОМАНД. ИХ УДАЛЕНИЕ С ПОСЛЕДУЮЩЕЙ ПЕРЕЗАГРУЗКОЙ ПРИВОДИТ К ТОМУ, ЧТО В ТЕЧЕНИЕ НЕСКОЛЬКИХ ТАКТОВ КОНВЕЙЕР НЕ ПОКИДАЕТ НИ ОДНА КОМАНДА. ПРОЦЕССОР, В КОТОРОМ ЭТО СЛУЧАЕТСЯ ЧАСТО, ТЕРЯЕТ 20... 30 % СВОЕЙ ПРОИЗВОДИТЕЛЬНОСТИ. ПОЭТОМУ RISC-ПРОЦЕССОРЫ ХАРАКТЕРИЗУЮТСЯ ВЕСЬМА ЭФФЕКТИВНЫМИ МЕХАНИЗМАМИ ПРЕДСКАЗАНИЯ ВЕТВЛЕНИЙ. КРОМЕ ТОГО, ОНИ СОДЕРЖАТ УСТРОЙСТВА, ПОЗВОЛЯЮЩИЕ ВЫБРАТЬ ТЕ ИЗ КОМАНД В ПРЕДСКАЗАННОМ ПЕРЕХОДЕ, КОТОРЫЕ МОЖНО ВЫПОЛНИТЬ ПРЕЖДЕ, ЧЕМ СТАНЕТ ЯСНО, ПРАВИЛЬНО ЛИ БЫЛ ПРЕДСКАЗАН ПЕРЕХОД.

ЕЩЕ ОДНА ОСОБЕННОСТЬ RISC-ПРОЦЕССОРОВ - ИСПОЛЬЗОВАНИЕ БОЛЬШОГО ЧИСЛА РЕГИСТРОВ. КАК ПРАВИЛО, В RISC-ПРОЦЕССОРАХ ИХ НЕ МЕНЕЕ 32. ПОДОБНАЯ СВОБОДА (СЕМЕЙСТВО X86 ИМЕЕТ ВСЕГО ВОСЕМЬ РЕГИСТРОВ ОБЩЕГО НАЗНАЧЕНИЯ) ПОЗВОЛЯЕТ СНИЗИТЬ ЧИСЛО ОБРАЩЕНИЙ К ОТНОСИТЕЛЬНО МЕДЛЕННОЙ ОПЕРАТИВНОЙ ПАМЯТИ В ПОЛТОРА-ДВА РАЗА (В СРАВНЕНИИ С CISC-ПРОЦЕССОРАМИ), ЧТО ОПЯТЬ-ТАКИ ПОЛОЖИТЕЛЬНО СКАЗЫВАЕТСЯ НА РОСТЕ РЕАЛЬНОЙ

ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ. ДОБАВИМ К ЭТОМУ, ЧТО ВСЕ RISC-ПРОЦЕССОРЫ СОДЕРЖАТ СИСТЕМЫ УПРАВЛЕНИЯ КЭШ-ПАМЯТЬЮ ВТОРОГО УРОВНЯ, ПОЗВОЛЯЮЩИЕ РАБОТАТЬ С НЕЙ НА МАКСИМАЛЬНОЙ СКОРОСТИ (В RISC-ИЗДЕЛИЯХ ВЗАИМОДЕЙСТВИЕ С КЭШ-ПАМЯТЬЮ ОБЫЧНО ПРОИСХОДИТ НА ЧАСТОТЕ, БОЛЕЕ НИЗКОЙ, ЧЕМ ЧАСТОТА САМОГО ПРОЦЕССОРА).

Очевидно, что RISC-процессоры эффективны в тех областях применения, в которых можно продуктивно использовать структурные способы уменьшения времени доступа к оперативной памяти. Если программа генерирует произвольные последовательности адресов обращения к памяти и каждая единица данных используется только для выполнения одной команды, то фактически производительность процессора определяется временем обращения к основной памяти. В этом случае использование сокращенного набора команд только ухудшает эффективность, так как требует пересылки операндов между памятью и регистром вместо выполнения команд "память, память - память". Программист должен учитывать необходимость локального размещения обрабатываемых данных, чтобы при пересылках между уровнями памяти по возможности все данные пересылаемых блоков данных принимали участие в обработке. Если программа будет написана так, что данные будут размещены хаотично и из каждого пересылаемого блока данных будет использоваться только небольшая их часть, то скорость обработки замедлится в несколько раз до скорости работы основной памяти.

Развитие микропроцессоров происходит при постоянном стремлении сохранения преемственности программного обеспечения (ПО) и повышения производительности за счет совершенствования архитектуры и увеличения тактовой частоты. Сохранение преемственности ПО и повышение производительности, вообще говоря, противоречат друг другу. Процессоры с системой команд x86, относящиеся к классу CISC-процессоров, имеют более низкие тактовые частоты по сравнению с микропроцессорами ведущих компаний-изготовителей RISC-процессоров. Существуют приложения, на которых производительность x86 микропроцессоров значительно ниже, чем у RISC-процессоров, реализованных на той же элементной базе. Однако возможность использования совместимого ПО для различных поколений x86 процессоров, выпущенных в течение последнего десятилетия, обеспечивает им устойчивое доминирующее положение на рынке.

В настоящее время на основе разработок компаний NexGen и AMD, подхваченных компанией Intel, предпринята попытка решить проблему повышения производительности в рамках архитектуры x86. Эти компании, сохраняя преемственность по системе команд с CISC-микропроцессорами семейства x86, создают новые устройства с использованием элементов RISC-архитектуры, использующие концепцию разделенной (decoupled) архитектуры и RISC ядра. В микропроцессор встраивается аппаратный транслятор, превращающий команды x86, в команды RISC-процессора. При этом одна команда x86 может породить до четырех команд RISC-процессора. Исполнение команд происходит как в развитом суперскалярном процессоре.

Лекция 5.

Методы ускорения переключения контекста процессора

Современные операционные системы и системы программирования широко используют переключение контекста процессора (содержимого регистров и отдельных управляющих триггеров) при отработке входа в прерывание и выхода из него, входа и выхода из подпрограммы и в случае организации мультипрограммной работы. Время переключения контекста должно быть по возможности минимальным, так как затраты на переключение - это плата за организацию совместного протекания совокупности взаимодействующих вычислительных процессов.

Уменьшение времени переключения контекста процессора может быть достигнуто за счет, во-первых, сокращения количества регистров, содержимое которых сохраняется в памяти, во-вторых, аппаратной поддержки сохранения регистров и, в-третьих, введения специальных соглашений, регламентирующих использование регистров в программах, что позволяет перейти от полного сохранения контекста к частичному.

Уменьшение количества сохраняемых регистров ведет к снижению производительности и, вообще говоря, находится в противоречии со стремлением увеличения производительности за счет использования быстрой регистровой памяти и параллельного функционирования устройств процессора, каждое из которых содержит собственные регистры. Однако, этот прием применяется в ряде архитектур, например в транспьютерах компании INMOS и Java-процессорах. Это архитектуры, основанные на операциях со стеком. К числу сохраняемых регистров относятся указатели на текущую позицию стека, на используемую область памяти и т.д. Число таких указателей ограничено, например, в транспьютере их 8, что позволило переключать контекст за 2 микросекунды при тактовой частоте 10 МГц.

Аппаратная поддержка сохранения регистров может реализовываться по разному. С одной стороны, это может быть ускоренный аппаратный перенос содержимого регистров в память. С другой - возможно предоставление каждой вновь активизируемой программе своего множества регистров.

Например, в архитектуре SPARC микропроцессоров компании SUN используется 200 регистров, образующих 8 групп (окон) по 32 регистра с общими для двух соседних окон восемью регистрами. Перекрытие регистровых окон выполнено так, что регистры с номерами 24-31 предыдущего окна служат одновременно регистрами с номерами 0-7 последующего окна. Это, по мнению разработчиков, увеличивает эффективность передачи параметров подпрограммам.

Использование регистров обработки быстрых прерываний. Ряд функций, связанных с обработкой прерываний, выполняются специализированными программами, которые могут использовать ограниченное число регистров. Поэтому в ряде процессоров вводятся специальные регистры, используемые при обработке так называемых быстрых прерываний. Это, например,

прерывание по приему очередного символа сообщения, переносящее символ в память. Действия по началу приема, требующие запуска механизма распределения памяти, и завершению приема всего сообщения требуют, как правило, обычного прерывания, сохраняющего все регистры процессора. Но так как чаще происходит прием очередного символа, то быстрые прерывания дают существенное увеличение производительности.

Стандартизация архитектур микропроцессоров

На протяжении всей истории развития вычислительной техники предпринимались попытки (прежде всего со стороны разработчиков программных средств) стандартизировать архитектуру процессоров, что существенно расширило бы область применения создаваемого программного обеспечения. Осознав безуспешность попыток добиться совместимости на уровне системы машинных команд, разработчики пытались стандартизовать язык ассемблера, языки высокого уровня, языки интерфейса прикладных программ с операционными системами.

Стимулом к этому была и остается постоянно растущая сложность как самих процессоров, так и создаваемых с их использованием программных систем.

Создание сложных новых систем требует, помимо всего прочего, наличия двух обязательных этапов: адекватного описания системы и исчерпывающего тестирования на соответствие этому описанию. Тестирование должно быть доказательным. Не прибегая к примерам из области создания больших прикладных систем, укажем на широко известные ошибки в микропроцессорах известных компаний и на наличие не декларированных возможностей микропроцессоров и операционных систем.

Отсутствие стандартизации не позволяет создавать новые системы путем конструирования из существующих, прошедших апробацию в разнообразных условиях применения большим количеством независимых пользователей.

Попытка комплексного решения проблемы стандартизации - формулирование концепции Открытых систем. Открытые системы представляют совокупность интерфейсов, протоколов и форматов данных, базирующихся на общедоступных, общепринятых стандартах, обеспечивающих переносимость (мобильность) программного обеспечения, взаимодействие между системами, масштабируемость.

Переносимость - свойство, выражающееся в возможности исполнения программы в исходных кодах на различных аппаратных платформах в среде различных операционных систем.

Взаимодействие систем - свойство, выражающееся в способности систем обмениваться информацией с автоматическим восприятием форматов и семантики данных.

Масштабируемость - свойство, выражающееся в возможности исполнения программы на различных ресурсах (объем памяти, число и производительность процессоров) с пропорциональным изменению ресурсов значением показателей

эффективности. Важно понимать, что ресурсы могут не только возрастать, но и уменьшаться. Например, программа может выполняться на произвольном, выделенном для ее исполнения участке памяти.

В рамках концепции Открытых систем архитектура процессора должна поддаваться достаточно простому формальному описанию со спецификацией типов данных, регистров и выполняемых преобразований без "побочных эффектов".

Известны по крайней мере две попытки реализации этого подхода, рассмотренные ниже.

Архитектурно независимая спецификация программ. В настоящее время в рамках международной организации ISO/IEC в комитете по микропроцессорным системам ведется подготовка проекта стандарта ANDF на архитектурно независимый формат спецификации программ (Architecture Neutral Distribution Format). По мнению разработчика компании X/Open Company Ltd., этот формат спецификаций позволит решить проблему переносимости программ. Компиляция исходного кода предполагается двухэтапной. На первом этапе исходный код транслируется в обобщенные декларации интерфейсов прикладных программ (API) в совокупности с обобщенными описаниями типов данных. Фактически полученная оттранслированная программа представляет собой выражение абстрактной алгебры, определенной Architecture Neutral Distribution Format. В результате текст программы может быть подвергнут формальной проверке и преобразованию. На втором этапе генерируется программа для конкретной архитектуры.

Java-технология, предложенная компанией SUN. В основе данной технологии лежит понятие виртуальной Java-машины, спецификации которой включают следующие типы данных:

- byte - байт;
- short - двухбайтовое целое;
- integer - четырехбайтовое целое;
- long - восьмибайтовое вещественное;
- float - четырехбайтовое вещественное;
- double - восьмибайтовое вещественное;
- char - двухбайтовый символ;
- object - четырехбайтовая ссылка на объект;
- returnAddress - четырехбайтовый адрес возврата. В виртуальном Java-процессоре предусмотрены следующие регистры:
 - PC - счетчик команд;
 - Vars - регистр для доступа к локальным переменным;
 - Optr - указатель на стек операндов;
 - Frame - указатель на окружение времени выполнения. Большинство команд

Java-процессора имеют длину один байт, что согласуется со стековой архитектурой процессора, использующей небольшое число регистров и указателей на данные,

Использование байт-кода в процессоре Java позволяет уменьшить длину программ. Средняя длина команды составляет 1,8 байта. В последнее время ко всем ранее существовавшим доводам в пользу стандартизации архитектур добавилась практическая потребность работы в сетях типа Internet, что выдвигает требование короткого программного кода. Открытые системы, создаваемые в Internet, позволяют накапливать программные продукты и конструировать системы из уже существующих.

Заметим, что архитектура виртуальной Java-машины достаточно похожа на архитектуру транспьютеров компании IMMOS. Отличие фактически состоит в добавлении элементов объектно-ориентированной технологии. Одним из препятствий на пути развития Java-технологии является низкая производительность исполнения Java-кода. Однако есть все предпосылки для преодоления этого препятствия. Например, современные процессоры с архитектурой компании Intel x86 содержат специальный блок, транслирующий сложные команды в совокупность простых команд RISC-процессора. Далее RISC-процессор исполняет эти команды, используя все преимущества RISC-подхода для достижения высокой производительности. Вполне мыслимое дело разработать подобный транслятор для Java-кода, когда байт-код будет транслироваться в команды реального процессора.

Другим возможным подходом к повышению производительности служит примененный в транспьютере T-9000. В нем предпринята попытка при сохранении байтовой системы команд транспьютеров семейств T-2xx, T-4xx, T-8xx повысить скорость исполнения за счет одновременной обработки большого числа команд при исполнении их на параллельно функционирующих обрабатывающих устройствах.

Лекция 6.

ILP процессоры

Архитектура ILP- процессоров

Более высокая производительность достигается как за счёт совершенствования полупроводниковой технологии, так и за счёт увеличения плотности микросхем. Дальнейшего увеличения скорости выполнения программ можно добиться в первую очередь благодаря реализации определённого вида параллелизма. Параллелизм на уровне команд (instruction-level parallelism, ILP) стал возможен благодаря созданию процессоров и методик компиляции, которые ускоряют работу за счёт параллельного выполнения отдельных RISC-операций. Системы на базе ILP используют программы, написанные на традиционных языках высокого уровня для последовательных процессоров, а обнаружение «скрытого параллелизма» автоматически выполняется благодаря применению соответствующей

компиляторной технологии и аппаратного обеспечения.

Тот факт, что эти методики не требуют от прикладных программистов дополнительных усилий, имеет крайне важное значение. Это решение резко отличается от традиционного микропроцессорного параллелизма (многопроцессорный и мультискалярный тип параллельной обработки), который предполагает, что программисты должны переписывать свои приложения. Поэтому сейчас, параллельная обработка на уровне команд, является единственным надёжным подходом, позволяющим добиться увеличения производительности без фундаментальной переработки приложений. Эти два типа параллельной обработки не исключают друг друга; самые эффективные многопроцессорные или мультискалярные системы, вероятнее всего, будут создаваться на базе процессоров ILP.

Компьютерная архитектура – это своего рода соглашение между классом программ, написанных для данной архитектуры, множеством реализаций процессора для неё. Как правило, это соглашение описывает формат и интерпретацию отдельных команд, но в случае с архитектурами ILP это соглашение может быть расширено: в него включается информация о возможном параллелизме между командами.

Есть два крайних подхода, при возможных промежуточных, к отображению присущего микропроцессору внутреннего параллелизма обработки данных на архитектурном уровне в системе команд. Первый подход более консервативен и состоит в том, что никакого указания на параллельную обработку внутри процессора система команд не содержит. Такие процессоры относятся к классу суперскалярных.

Второй подход - напротив полностью открывает все возможности параллельной обработки. В специально отведенных полях команды каждому из параллельно работающих обрабатывающих устройств предписывается действие, которое устройство должно совершить. Такие процессоры называются процессорами с длинным командным словом (VLIW). Предполагается, что существуют компиляторы с языков высокого уровня, которые готовят программы для загрузки их в микропроцессоры.

Основная идея, определяющая развитие ILP микропроцессоров, состоит в построении возможно большего количества параллельных структур при сохранении традиционных последовательных программ. Это означает, что компиляторы и аппаратура микропроцессора сами, без вмешательства программиста, обеспечивают загрузку параллельно работающих функциональных устройств микропроцессора.

Повысить степень параллелизма программы можно изменяя соответствующим образом ее статическую или динамическую структуру. Поскольку статическая структура программы однозначно соответствует ее исходному тексту (в предположении неизменности компилятора), то изменение статической структуры сводится к изменению исходного кода, что, в общем случае, не всегда возможно. Динамическая же структура программы может быть изменена при неизменной статической структуре. И главной целью такого

изменения должно быть повышение степени параллельного исполнения команд.

Суперскалярные процессоры

Суперскалярные процессоры – это реализация ILP-процессора для последовательных архитектур – архитектур, программа для которых не должна передавать и, фактически, не может передавать точную информацию о параллелизме. Поскольку программа не содержит точной информации о наличии ILP, то, задача обнаружения параллелизма должна решаться аппаратурой, которая, следовательно, должна создавать план действий для обнаружения «скрытого параллелизма». Код для суперскалярных процессоров содержит последовательность команд, которая порождает корректный результат, если выполняется в установленном порядке. Код указывает последовательный алгоритм и, за исключением того, что он использует конкретный набор команд, не представляет себе точную природу аппаратного обеспечения, на котором он будет работать или точный временной порядок, в котором будут выполняться команды. Такой подход увеличивает сложность аппаратного обеспечения, в то же время суперскалярный процессор создаёт план выполнения, используя преимущества тех факторов, которые могут быть определены только во время выполнения.

Допустимые границы преобразования динамической структуры программы задают существующие на множестве инструкций отношения: зависимость по управлению и зависимость по данным. При описании архитектур суперскалярных процессоров часто используется модель окна исполнения. При исполнении программы микропроцессор как бы продвигает по статической структуре программы окно исполнения. Команды в окне могут исполняться параллельно, если между ними нет зависимости.

Для устранения зависимостей, вызванных командами переходов, используется метод предсказания, позволяющий извлекать и условно исполнять команды предсказанного перехода. Если позднее обнаруживается, что предсказание было сделано верно, то результаты условно исполненных команд принимаются. Если предсказание было ошибочным, состояние процессора восстанавливается на момент принятия решения о выполнении перехода.

Команды, помещенные в окно исполнения, могут быть зависимы по данным. Эти зависимости обусловлены использованием одних и тех же ресурсов памяти (регистров, ячеек памяти) в разных командах. Поэтому для правильного исполнения программы необходимо использование этих ресурсов в предписываемом программой порядке.

Все виды зависимостей по данным могут быть классифицированы по типу ассоциаций: RAR - "чтение после чтения", WAR - "запись после чтения" и WAW - "запись после записи", RAW - "чтение после записи". Пример различных зависимостей команд по данным показан на рис.11. Некоторые из зависимостей по данным могут быть устранены. RAR, по сути дела,

соответствует отсутствию зависимостей, поскольку в данном случае порядок выполнения команд не имеет значения. Действительной зависимостью является только "чтение после записи" (RAW), так как необходимо прочитать предварительно записанные новые данные, а не старые.

Лишние зависимости по данным появляются в результате "записи после чтения" (WAR) и "записи после записи" (WAW).

После удаления лишних зависимостей по управлению и данным команды могут исполняться параллельно. Формирование расписания параллельного выполнения команд возлагается на аппаратные средства микропроцессора. Это расписание учитывает существующие зависимости между командами и имеющиеся функциональные модули процессора.

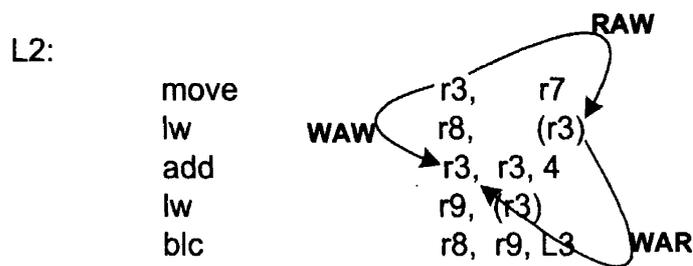


Рис.11. Зависимости команд по данным

Типовая архитектура суперскалярного микропроцессора представлена на рис.12.

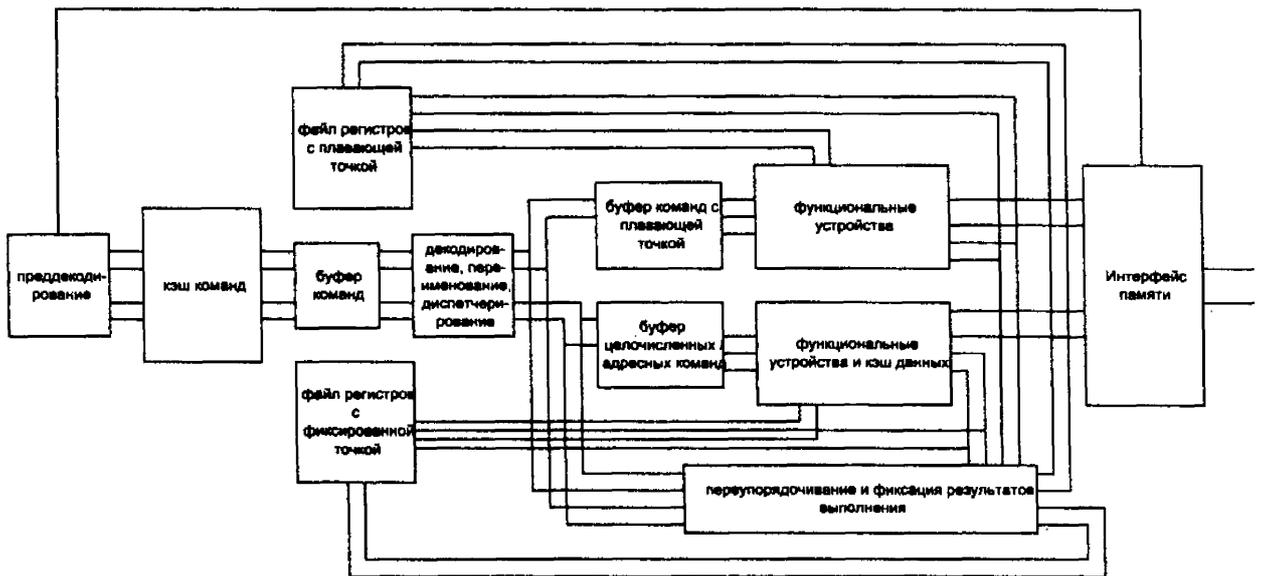


Рис.12. Архитектура суперскалярного микропроцессора

В число основных блоков суперскалярного микропроцессора входят: блок выборки команд и предсказания переходов, блок декодирования команд, анализа зависимостей между командами, переименования и диспетчеризации, блоки регистров и обрабатывающих устройств с плавающей и фиксированной

точками, блок управления памятью, а также блок упорядочения выполненных команд.

Рассмотрим основные приемы повышения быстродействия в суперскалярных микропроцессорах.

Предварительная выборка команд и предсказание переходов. Поскольку при суперскалярной обработке необходимо извлекать из памяти несколько команд за один такт для загрузки параллельно работающих функциональных модулей, повышенные требования предъявляются к пропускной способности интерфейса микропроцессор-память. В современных микропроцессорах применяются многоуровневые отдельные кэш-памяти данных и команд.

Для уменьшения потерь процессорных циклов, связанных с промахами при обращении к кэш-памяти в случае выполнения команд ветвления, в состав системы кэширования введены средства предсказания переходов, основное назначение которых - повысить вероятность наличия в кэш-памяти требуемой команды.

Исполнение условных ветвлений состоит из следующих этапов:

- распознавание команды условного ветвления;
- проверка выполнения условия перехода;
- вычисление адреса перехода;
- передача управления, в случае перехода.

На каждом этапе используются специальные приемы повышения производительности:

1. Для быстрого декодирования используются либо дополнительные биты в поле команды, либо преддекодирование команд при выборе из кэш-памяти команд.

2. Часто, когда команда уже выбрана в кэш, условие перехода еще не вычислено и чтобы не задерживать поток команд в данном случае используется предсказание перехода по одной из нескольких возможных схем. Некоторые предсказатели используют статическую информацию из двоичного кода программы или специально выработанную компилятором. Например, определенные коды операций чаще вырабатывают ветвление, чем другие коды, или ветвление более вероятно (при организации циклов), или компилятор может устанавливать флаг, указывающий направления перехода. Может также использоваться статистическая информация, полученная при трассировке программы.

Другие предсказатели используют динамически формируемую информацию в процессе исполнения программы. Обычно это информация, касающаяся истории выполнения данного ветвления, сохраняемая в таблице ветвлений или в таблице предсказаний ветвлений. Таблица предсказания ветвлений организуется по ассоциативному принципу, подобно кэш-памяти, ее элементы доступны по адресу команды, ветвление которой предсказывается. В некоторых реализациях элемент таблицы предсказания ветвления является счетчиком,

значение которого увеличивается при правильном предсказании и уменьшается при неправильном. При этом значение счетчика определяет преобладающее направление ветвлений.

В момент определения действительного значения условия ветвления, вносится изменение в историю ветвления. Если предсказание было неверным, то должна иницироваться выборка правильных команд. Результаты команд, которые были условно выполнены, должны быть аннулированы.

3. Для определения адреса ветвления обычно требуется выполнить целочисленное сложение, прибавляющее к текущему значению счетчика команд смещение, заданное в поле команды ветвления. И хотя это не требует дополнительных циклов для обращения к регистрам, ускорение вычисления адреса может быть достигнуто благодаря использованию буфера, содержащего ранее использованные адреса переходов.

Если требуется осуществить смену значения счетчика команд, то необходимо, по крайней мере, один такт для распознавания команды ветвления, модификации счетчика команд и выборки команды по заданному значению счетчика команд. Эти задержки вызывают пустые такты в конвейерах процессора. Более сложные решения используют буферы, содержащие наборы команд для двух возможных результатов ветвлений.

Возможно также использование "отложенных переходов", когда одна или несколько команд после команды ветвления выполняются безусловно.

Декодирование команд, переименование ресурсов и диспетчеризация. На этой фазе определяются существенные зависимости (RAW) по данным между командами и преодолеваются несущественные (WAW, WAR), производится распределение команд по буферам команд функциональных устройств.

При декодировании команды создается одна или несколько упорядоченных троек, каждая из которых включает: 1) исполняемую операцию, 2) указатели на операнды, 3) указатель на место помещения результата.

Для преодоления лишних WAR и WAW зависимостей, возникающих в результате ограниченности логических ресурсов (ячеек памяти, регистров), используется механизм динамического отображения определяемых текстом программы логических ресурсов на физические ресурсы микропроцессора. При данном подходе с одним логическим ресурсом может быть связано несколько значений в различных физических ресурсах, каждое из которых соответствует значению логической величины в один из моментов времени последовательного выполнения программы.

Когда команда создает новое значение для логического регистра, физический ресурс, в который помещается это значение, получает имя. Последующие команды, использующие это значение, снабжаются именем физического ресурса. Данная процедура называется переименованием регистров. Используются два основных способа переименования.

В первом, физический файл регистров больше логического. При необходимости переименования из списка свободных физических регистров

берется один и ему сопоставляется соответствующее логическое имя. Если список свободных регистров пуст, диспетчеризация команд приостанавливается до момента появления свободных физических регистров.

Остается вопрос о возвращении физических регистров в список свободных после того, как из них считаны данные в последний раз. Один из способов связывает счетчик с каждым физическим регистром. Счетчик увеличивается при каждом переименовании операнда в командах, использующих этот физический регистр. Соответственно при использовании операнда значения счетчика уменьшается на 1. При достижении счетчиком нуля физический ресурс должен быть переведен в список свободных.

Второй способ переименования использует одинаковое число логических и физических регистров и поддерживает их однозначное соответствие. В дополнение имеется буфер с одним входением для каждой инициированной на исполнение команды. Этот буфер называется переупорядочивающим, так как он используется также для установления порядка команд при прерываниях. Данный буфер можно рассматривать в виде кольцевого буфера с указателями "начало" и "конец".

Команды помещаются в конец буфера. По завершению команды ее результат заносится в заранее предписанный ей элемент очереди, независимо от места в очереди, занимаемого этим элементом. К моменту достижения командой начала буфера, если она была исполнена, ее результат помещается в регистровый файл, а сама команда удаляется. Команда, находящаяся в буфере и не исполненная в виду отсутствия значения операнда, остается в нем вплоть до получения этого значения. Одновременно может выбираться из очереди или помещаться в нее несколько команд.

Исполнение команд. После формирования для каждой команды упорядоченных троек, состоящих из кода операции, физических операндов - источника и результата, и размещения их в буферах, наступает фаза динамической проверки готовности значений операндов для исполнения команды.

В идеале команда готова к исполнению как только готовы ее входные операнды. Однако есть ряд ограничений, связанных с доступностью физических ресурсов, таких как исполнительные устройства, коммутаторы и порты регистровых файлов (или переупорядочивающего буфера). Для организации окна исполнения используются различные методы: одной очереди, многих очередей или метод резервирующей станции.

Если имеется одна очередь, то переименование регистров не требуется, так как доступность значений операндов может отмечаться битом резервирования, сопоставленным каждому регистру. Регистр резервируется, когда модифицирующая его команда назначается на исполнение. И регистр освобождается, когда заканчивается исполнение команды. Если для команды ресурсы не были зарезервированы, то она приостанавливает свое исполнение.

В методе многих очередей каждая очередь организуется для команд одного

типа. Например, очередь команд с плавающей точкой или очередь команд работы с памятью.

Третий метод предполагает использование резервирующей станции, состоящей из совокупности элементов, каждый из которых содержит позиции для размещения кода операции, наименования первого операнда, самого первого операнда, признака доступности первого операнда, наименования второго операнда, самого второго операнда, признака доступности второго операнда и наименования регистра результата. Когда команда завершает исполнение и вырабатывает результат, то наименование результата сравнивается с наименованиями операндов в резервирующей станции.

Если в резервирующей станции обнаруживается команда, ждущая этого результата, то данные записываются в соответствующую позицию и устанавливается признак их доступности. Когда у команды доступны все операнды, инициируется ее исполнение. Резервирующая станция следит за доступностью операндов. Когда команда при диспетчеризации попадает в резервирующую станцию, все готовые операнды из регистрационного файла переписываются в поля этой команды. Когда все операнды готовы, команда исполняется. Иногда резервирующая станция содержит не сами операнды, а указатели на них в регистрационном файле или переупорядочивающем буфере.

Работа с памятью. Для вычисления адреса памяти, как правило, требуется, по крайней мере, одно сложение. После вычисления адреса может потребоваться его преобразование в физический адрес, осуществляемое буфером трансляции адресов (TLB).

Проблемы конфликтов при доступе к разделяемому ресурсу - ячейкам памяти, по сути те же, что и при доступе к регистрам.

Завершение выполнения команды. Завершающей фазой исполнения команды является фаза изменения состояния процессора в соответствии с выполненной командой. Назначение этой фазы - сохранение последовательной модели исполнения программы, при реальном параллельном выполнении отдельных команд и условном выполнении команд ветвления. Для изменения состояния процессора применяются два основных способа, причем оба основаны на использовании двух состояний: состояния, измененного в результате операции, и состояния, требуемого для восстановления.

При первом способе сохраняется состояние процессора в наборе контрольных точек или в буфере истории вычислений, которые, в случае необходимости, используются для восстановления состояния.

Второй способ предполагает рассмотрение логического (архитектурного) и физического состояния процессора. Физическое состояние изменяется немедленно по завершении очередной команды. Архитектурное состояние изменяется тогда, когда ясен результат условно выполненных команд. Для реализации этого способа используется переупорядочивающий буфер: результаты из буфера отправляются в файл архитектурных регистров и память.

В переупорядочивающем буфере для каждой команды содержится

соответствующее ей значение счетчика команд и значения других регистров, которые необходимы для корректного обслуживания прерываний.

На рис.13. показаны основные компоненты суперскалярного микропроцессора: функциональные модули - выполнения операций с плавающей (FPU) и фиксированной (ALU) точкой, устройство загрузки/сохранения, файлы регистров, отдельная кэш-память команд и данных, а также вспомогательные модули, обеспечивающие динамическое планирование вычислительного процесса " устройство связи с кэш-памятью 2-го уровня, блок переупорядочивания команд и блок предварительной дешифрации.

По крайней мере два обстоятельства ограничивают эффективность использования суперскалярных архитектур. Во-первых, есть ограничения на степень параллелизма на уровне команд, даже если применяется самая совершенная техника суперскалярных вычислений. Первое ограничение проистекает из условных переходов. Другое следует из того, что размер окна исполнения (число активных команд, могущих исполняться параллельно) ограничивает возможный присущий программе параллелизм, так как не рассматривается параллельное исполнение команд, находящихся на расстоянии, превышающем размер окна.

Во-вторых, сложность суперскалярного процессора возрастает как количество параллельно исполняемых команд и даже быстрее.

Вероятнее всего, что пределом распараллеливания при суперскалярной обработке является запуск одновременно на исполнение в каждом такте 7-8 команд.

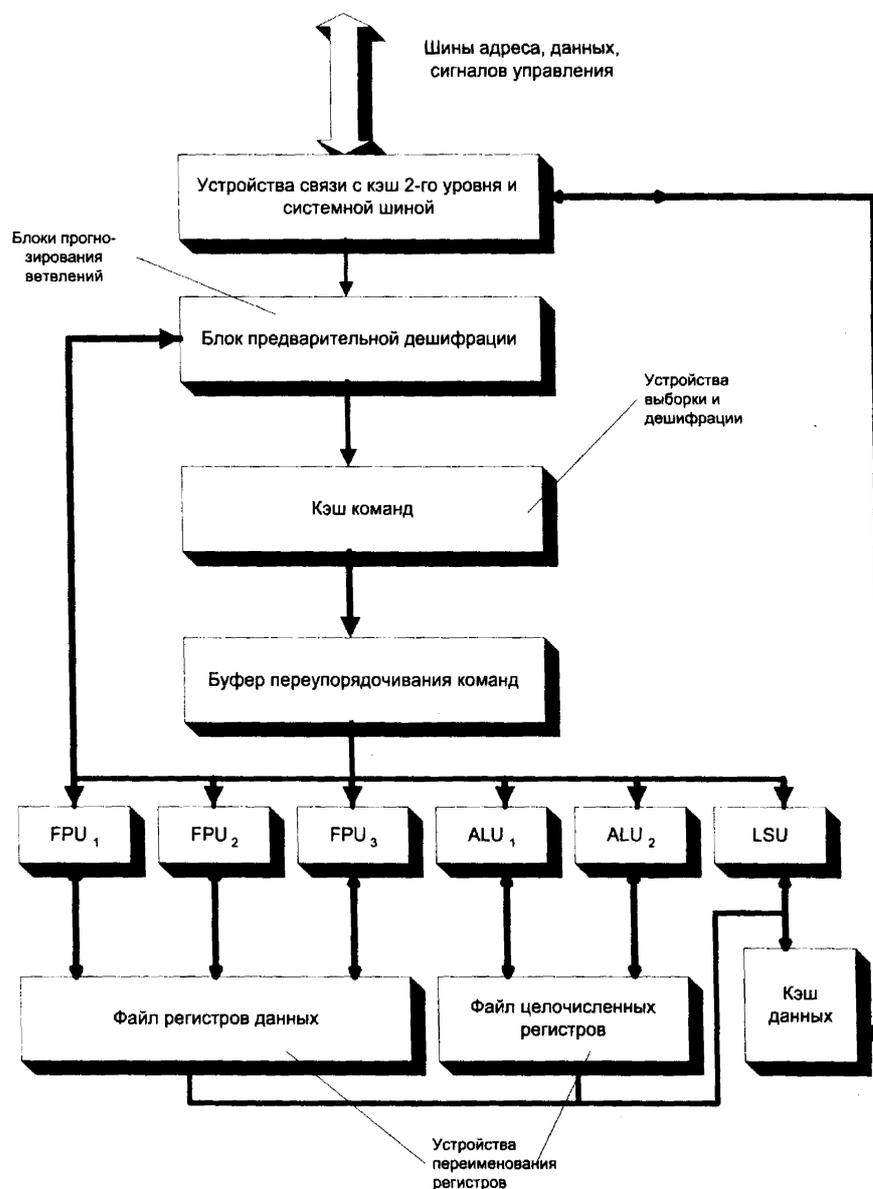


Рис. 13. Структура суперскалярного микропроцессора

VLIW-процессоры.

Альтернатива суперскалярной обработке – процессоры с длинным командным словом (VLIW). Использование этого метода предполагает задание в командном слове совокупности параллельно выполняемых команд. Подготовкой таких программ занимается компилятор.

В отличие от программ для суперскалярных процессоров, код VLIW предлагает точный план того, как процессор будет выполнять программу, план, которой компилятор создаёт статически во время компиляции. Код точно указывает, когда будет выполнена каждая операция, какие функциональные устройства будут работать и какие регистры будут содержать операнды. Компилятор VLIW создаёт такой план выполнения, имея полное представление о процессоре VLIW, причём создаёт этот план так, чтобы добиться требуемой

записи выполнения – последовательности событий, которые действительно происходят во время работы программы. Компилятор передаёт план выполнения (через архитектуру набора команд, которая точно описывает параллелизм) аппаратному обеспечению, которое, в свою очередь, выполняет этот план.

Процессоры VLIW представляют собой пример архитектуры, для которой программа представляет точную информацию о параллелизме. Компилятор выявляет параллелизм в программе и сообщает программному обеспечению, какие операции не зависят друг от друга. Эта информация имеет важное значение для аппаратного обеспечения, поскольку в этом случае оно «знает» без дальнейших проверок, какие операции можно начинать выполнять в одном и том же такте.

Достоинства VLIW заключаются в следующем:

Компилятор может эффективнее исследовать зависимости между командами и выбирать параллельно исполняемые команды, чем это делает аппаратура суперскалярного процессора, ограниченная размером окна исполнения.

VLIW процессор имеет более простое устройство управления и потенциально может иметь более высокую тактовую частоту.

Однако у VLIW процессоров есть серьёзный фактор, снижающий их производительность. Это команды ветвления, зависящие от данных, значения которых становятся известны только в динамике вычислений. Окно исполнения VLIW-процессора, не может быть очень большим в виду отсутствия у компилятора информации о зависимостях, формируемых динамически, в процессе выполнения. Этот недостаток препятствует возможности переупорядочивания операций в VLIW процессоре. Кроме того, VLIW реализация требует большого размера памяти имён, многовходовых регистровых файлов, большого числа перекрёстных связей.

EPIC-процессоры.

Тип архитектуры Explicitly Parallel Instruction Computing – это эволюция архитектуры VLIW, которая абсорбировала в себе многие концепции суперскалярной архитектуры, хотя и в форме, адаптированной к EPIC. По сути, EPIC – это «идеология», определяющая, как создавать ILP-процессоры, а также набор характеристик архитектуры, которые поддерживают данную идеологию. К архитектуре EPIC можно отнести множество различных архитектур набор команд (ISA). Помимо включения или исключения той или иной характеристики архитектуры, которые мы рассматриваем, проектировщики процессоров должны принимать традиционные решения по таким вопросам, как набор кодов операций, диапазон поддерживаемых типов данных и количество используемых регистров. Любая архитектура, относящаяся к классу EPIC должна реализовывать общую идеологию EPIC. В зависимости от того, какие из характеристик EPIC использует архитектура ISA, она может быть оптимизирована для различных приложений, например, для систем общего назначения или встроенных устройств. Из чего следует, что тип архитектуры

EPIC должен быть архитектурой общего назначения, то есть архитектурой, способной достигать высокой степени параллелизма на уровне команд как в числовых, так и в скалярных приложениях.

Одна из целей создания архитектуры EPIC, состоит в том, чтобы сохранить реализованный в VLIW принцип статического создания плана выполнения, но в то же время обогатить его возможностями суперскалярного процессора, которые позволили новой архитектуре лучше учитывать динамические факторы, традиционно ограничивающие параллелизм, свойственный VLIW. Чтобы добиться этих целей, «идеология» EPIC была построена на следующих основных принципах:

Компилятор должен играть ключевую роль в создании плана выполнения, а архитектура должна обеспечивать поддержку необходимых для этого компонентов.

Архитектура должна обеспечивать функции, которые помогают компилятору в использовании статических ILP.

Архитектура должна обеспечивать механизм для передачи плана выполнения компилятора аппаратному обеспечению.

Направления развития ILP архитектуры.

Как уже отмечалось ранее, в ILP процессорах предпринимается попытка в рамках модели последовательных программ реализовать параллельное исполнение команд этих программ. После извлечения последовательного потока команд между командами устанавливаются только действительно необходимые зависимости по данным. При этом сохраняется достаточно информации о порядке следования команд в исходной программе, чтобы сохранить их порядок при наступлении прерывания.

Типичный ILP процессор выбирает команды и исследует их по мере выполнения. Исследование проводится с целью выявления и обработки команд перехода, идентификации типа команды для ее дальнейшего направления на соответствующий исполнительный блок или в буфер памяти. Выполняются также некоторые действия для смягчения зависимостей по данным, например переименование регистров. VLIW процессор возлагает на компилятор статическую реализацию тех функций, которые в суперскалярном процессоре выполняются динамически.

За последние десять лет сравнение достоинств VLIW и суперскалярных архитектур было основной темой в дискуссиях специалистов по вопросам ILP. Сторонники той и другой концепции сводят обсуждение к противопоставлению простоты и ограниченных возможностей VLIW и сложности и динамическим возможностям суперскалярных систем. При этом ясно, что оба подхода имеют свои достоинства и говорить об их альтернативности не уместно. Очевидно, что создание плана выполнения во время компиляции существенно для обеспечения высокой степени распараллеливания на уровне команд, даже для

суперскалярного процессора. Так же ясно и то, что во время компиляции существует неоднозначность, которую можно разрешить только во время выполнения, и для решения этой задачи процессор требует наличия динамических механизмов. Сторонники EISC согласны с обеими позициями. Различие только в том, что компилятор может управлять такими динамическими механизмами, применяя их выборочно там, где это возможно. Столь широкие функции управления дают компилятору возможность использовать правила управления этими механизмами более оптимально, чем это позволяют аппаратные возможности.

Основные принципы EISC, наряду с возможностями архитектуры, которые их поддерживают, обеспечивают средства определения ILP-архитектур и процессоров, позволяющих добиться более высокой степени ILP при меньшей сложности аппаратуры в самых разных прикладных областях. IA-64 – пример того, как принципы EISC могут применяться к вычислительным системам общего назначения – области, где совместимость кода, имеет критически важное значение. EISC даёт надежды на устойчивый рост производительности микропроцессоров общего назначения на наших приложениях без кардинального переписывания этих приложений. Однако можно быть уверенным в том, что EISC будет играть столь же важную роль и на рынке высокопроизводительных встроенных систем. В этой области более жёсткие требования к соотношению цена/производительность и при этом более низкие требования к совместимости на уровне объектных модулей, что заставляет использовать более настраиваемые архитектуры.

Дальнейшее повышение производительности микропроцессоров связывается в настоящее время со статическим и динамическим анализом кода с целью выявления резервов параллелизма уровня отдельных команд и программных сегментов с использованием информации, предоставляемой компилятором языка высокого уровня. Исследования в данном направлении привели к разработке мультискалярной архитектуры процессоров, которые являются дальнейшим развитием ILP архитектуры.

Лекция 7.

Мультискалярные процессоры

Мультискалярная модель выполнения программы.

Мультискалярные процессоры используют агрессивную парадигму выполнения кода с целью извлечения параллелизма уровня команд из последовательной программы, представленной на языке высокого уровня. В соответствии с данной парадигмой программа разбивается на совокупность задач с помощью программных и аппаратных средств. Задача – часть программы, выполнению которой соответствует непрерывная область динамической последовательности команд (например, часть базисного блока, базисный блок, множество базисных блоков, одиночная итерация цикла, полный цикл, обращение к функции, и т.д.). Задачи программы статически

разграничиваются аннотациями. Зависимости между операторами программы по управлению представляются как граф управляющих зависимостей (ГУЗ), в котором вершинами являются задачи, а дугами задается порядок их выполнения. Динамика выполнения программы может рассматриваться как обход ГУЗ программы. На каждом шаге обхода мультискалярный процессор назначает одну задачу на один из процессорных элементов (ПЭ) для выполнения, без учета фактического содержания задачи, и продолжает обход ГУЗ от рассматриваемой вершины до следующей.

Задача назначается для выполнения некоторому процессорному элементу, передачей ему начального значения программного счетчика. Множество инициированных таким образом задач выполняется параллельно на процессорных элементах, результатом чего является выполнение множества команд за один процессорный такт.

Каждый из процессорных элементов выбирает и выполняет команды, принадлежащие выделенной ему задаче. Значения разделяемых процессорными элементами регистров копируются в каждый ПЭ. Результат модификации содержимого регистров динамически направляется множеству параллельных ПЭ в соответствии с генерируемыми компилятором масками.

Доступ к памяти осуществляется спекулятивно (условно, по предположению) без знания последовательности предшествующих команд загрузки или сохранения. Обращение к данным осуществляется параллельно многим ПЭ, обработка приостанавливается только в случае истинной зависимости данных.

Пример архитектуры мультискалярного процессора показан на рис.14.

В общих чертах мультискалярный процессор можно рассматривать как параллельную вычислительную систему, состоящую из совокупности ПЭ с программой-планировщиком, которая назначает задачи на ПЭ. Как только задача назначена на ПЭ, он выбирает и выполняет команды задачи, пока она не завершится. Множество ПЭ, каждый с собственным внутренним механизмом последовательного выполнения команд, поддерживают выполнение множества задач. Команды, содержащиеся внутри динамического окна исполнения, ограничены первой командой в самой ранней выполняемой задаче и последней командой в последней выполняемой задаче. При условии, что каждая задача может содержать циклы и обращения к функциям, эффективный размер окна может быть чрезвычайно большим. Существенным является то, что не все команды внутри этого широкого диапазона одновременно рассматриваются для выполнения, а только ограниченный набор внутри каждого из ПЭ.

Важно заметить, что задачи, хотя и разделены на группы команд, но не являются независимыми. Так как задачи являются частями последовательного потока команд, то отношения по данным и управлению между индивидуальными командами должны поддерживаться в процессе выполнения. Ключевым вопросом в мультискалярной реализации является обеспечение связи по данным и управлению между параллельными процессорами. То есть как обеспечить выполнение последовательного обхода ГУЗ, если фактически

выполняется непоследовательный обход.

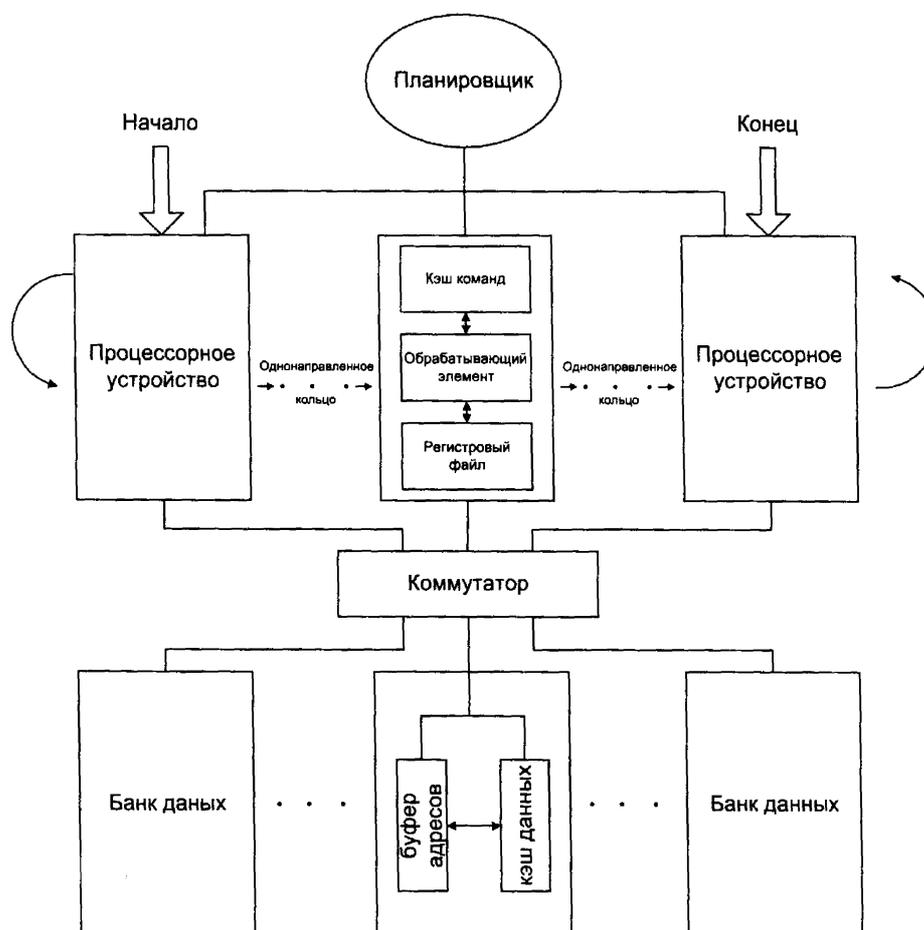


Рис 14. Микроархитектура мультискалярного процессора.

Последовательный обход поддерживается следующим образом. Во-первых, для каждого процессора обеспечивается последовательная модель выполнения назначенной ему задачи. Во-вторых, предписывается последовательный порядок выполнения для совокупности процессоров, который поддерживается с помощью организации циклической очереди ПЭ. Указатели начала и конца очереди идентифицируют ПЭ, которые выполняют самую раннюю и самую позднюю из назначенных текущих задач соответственно.

По мере выполнения команд задачи производятся и потребляются значения переменных программы. Эти значения связаны с местом хранения, а именно с регистрами и с памятью. Так как при последовательном выполнении область хранения переменных рассматривается как единый набор регистров и памяти, мультискалярное выполнение должно поддерживать такую же модель. Кроме того, мультискалярное выполнение должно гарантировать, что значения используются и производятся также, как и при последовательном выполнении. Чтобы обеспечить это, необходимо синхронизировать обмен между задачами.

В случае использования регистровой памяти логика управления

синхронизирует создание значений регистров в задачах-предшественниках с потреблением этих значений в задачах-преемниках. Производимые задачей регистровые значения могут быть определены статически и отмечены в маске создания задачи. В момент выработки соответствующего регистрового значения, если есть необходимая отметка в маске создания, это значение посылается через однонаправленный кольцевой канал (см. рис.14.) последующим задачам, т.е. в ПЭ, которые являются логическими преемниками ПЭ, выработавшего значение. Загружаемые из кольцевого канала в регистры значения, предназначенные для задач-преемников, определяются в маске накопления, которая является объединением масок создания активных в настоящее время задач-предшественников. Как только значения получены из модулей предшественников, очищаются признаки сохранения в модулях преемников. Если задача использует одно из этих значений, потребляющая команда может быть исполнена только в том случае, если значение было получено, иначе она ждет получения требуемого значения.

В отличие от значений регистров, для значений, хранимых в памяти, в силу динамического вычисления адресов, нельзя заранее точно определить, какие из них используются или производятся задачей. Если известно, что задача потребляет значение из памяти (используя команду загрузки), которое произведено (с помощью команды сохранения) в более ранней задаче, возможно синхронизировать потребление и производство этого значения. То есть загрузка в задаче-преемнице может быть отложена до тех пор, пока в задаче-предшественнице не будет выполнена команда сохранения (похоже на ситуацию с регистрами, однако механизм синхронизации все же другой, в виду несоизмеримости размеров пространства имен).

В более общем случае, когда такое знание недоступно, может быть предпринят консервативный или агрессивный подход. Консервативный подход подразумевает необходимость ожидания до тех пор, пока не возникнет уверенность, что команда загрузки прочтет правильное значение. Этот подход обычно подразумевает задержку выполнения команд загрузки внутри задачи, до тех пор пока не завершили операции записи в память все задачи-предшественницы, результат которых может быть использован последующей командой. При агрессивном подходе загрузки из памяти в регистры ПЭ должны выполняться спекулятивно, в предположении, что задача-предшественница позже не будет сохранять значения в ту же самую ячейку памяти. Чтобы гарантировать, что никакая задача-предшественница не записывает значение в ячейку памяти, предварительно считанную задачей-преемником, должна проводиться проверка в процессе выполнения вычислений. Если эта проверка идентифицирует загрузку и сохранение, которые находятся в противоречии (не происходят в соответствующем порядке), более поздняя задача должна быть прервана и должна быть инициализирована соответствующая процедура восстановления. В мультискалярных процессорах используется агрессивный подход.

Из-за спекулятивного характера мультискалярное выполнение должно иметь как средства подтверждения правильности выполнения, так и средства

исправления в случае неправильного выполнения. Выполнение команд внутри задач может рассматриваться как спекулятивное с двух точек зрения:

- спекулятивное по управлению;
- спекулятивное по данным.

Если в результате спекулятивного управления предсказание следующей задачи оказалось неверным, то следующая задача (задачи) должна быть отменена и восстановлена правильная последовательность задач. Аналогично задача, использующая неправильные данные, должна быть отменена и должно быть восстановлено правильное значение данных. В любом случае отмена задачи приводит к отмене всех задач, выполняемых после отмененной (иначе поддержание последовательной семантики оказывается сложным).

Для упрощения сохранения последовательной семантики исполнения программы мультискалярный процессор удаляет задачи из циклической очереди в том же порядке, в каком их помещал в очередь. В процессе спекулятивного выполнения задача производит значения, которые могут быть как правильными, так и неправильными. Только безусловно правильные результаты задачи могут быть безопасно использованы другими задачами. Тем не менее, в мультискалярном процессоре значения оптимистично посылаются для спекулятивного использования в процессе выполнения других задач. Поскольку задача посылает значения предварительно другим задачам, как только их вырабатывает, большая часть, если не все значения, будут посланы к моменту, когда задача становится головной в очереди. Таким образом, отмена задачи для освобождения процессора и назначения новой задачи может быть выполнена просто путем модифицирования указателя начала очереди.

Мультискалярные программы.

Мультискалярная программа должна обеспечить возможность быстрого обхода ее ГУЗ, в результате которого производится распределение множества задач по множеству процессоров.

Спецификация кода для каждой задачи стандартна. Задача определяется как фрагмент программы для последовательной машины. Хотя система команд, в которой представляется код, оказывает влияние на конструкцию каждого индивидуального процессора, это не оказывает влияния на остальную часть конструкции мультискалярного процессора.

Для ускорения обхода ГУЗ планировщику мультискалярного процессора требуется информация о структуре потока управления программы. В частности, требуется знать, какие задачи являются возможными преемниками любой данной задачи в ГУЗ. Планировщик мультискалярного процессора использует эту информацию для предсказания одной из возможных задач-преемников и продолжения обхода ГУЗ, начиная с текущей отметки. Такая информация может быть определена статически и помещена в описатель задачи. Описатели задачи могут быть расположены внутри текста программы (например, перед кодом задачи) или помещены отдельно, рядом с текстом программы (например,

в конце).

Для согласованного выполнения различных задач необходимо характеризовать каждую задачу в соответствии с набором используемых и производимых задачей значений.

Процедура обработки регистровых значений проста. В результате статического анализа ГУЗ компилятором формируется маска создания. Потребляя значения, задача ожидает их только в том случае, если они еще не были произведены задачей-предшественником. Иначе она находит требуемые значения внутри локальной памяти, переданные по кольцу задачей-предшественницей.

Естественным является расположение маски создания внутри описателя задачи. Так как задача может содержать множество базисных блоков, выполнение которых зависит от обрабатываемых данных, не представляется возможным определить статически, какие регистровые значения будут созданы динамике вычислений. Маска создания должна быть консервативной и вследствие этого включать все возможные регистровые значения, которые могут быть произведены.

По мере выполнения процессором команд задачи производимые значения регистров пересылаются последующим задачам. Так как ПЭ не может определять априорно, какие команды содержит назначенная ему задача, он не может знать, какие из команд выполняют модификацию регистров, чье значение должно быть послано другим задачам. В соответствии с последовательной семантикой другим задачам должен быть послан только результат последней модификации регистра в задаче. Стратегия, связанная с ожиданием выполнения всех команд в задаче (тогда никакие дальнейшие модификации регистров невозможны), нецелесообразна, так как это часто приводит к ожиданию другими задачами значения, которое уже является доступным.

Не все созданные задачей значения должны быть переданы задачам-приемникам. Достаточно передавать лишь те значения, которые будут использованы вне создавшей их задачи.

Компилятор имеет возможность определения последней команды в задаче, которая модифицирует соответствующий регистр. Он может отметить эту команду как специальную (выполнить-и-переслать) команду, которая в дополнение к выполнению определенной операции направляет результат следующим ПЭ. Кроме того, поскольку ПЭ выполняет команды задачи, он может идентифицировать те регистры, для которых значения не будут произведены.

По тем же самым причинам процессор не может определить, какие команды в действительности выполняет назначенная ему задача, так же как не может определять априорно, на какой команде задача завершится, т.е., в какой точке управление передается вне задачи. Во время разбиения компилятором ГУЗ на задачи определяются границы задачи и узлы передачи управления. Команда в

одном из этих узлов передачи управления может быть отмечена специальными условиями останова так, чтобы соответствующие условия могли быть оценены к моменту выборки такой команды процессором. Если связанные с командой условия останова выполнены, то задача завершена.

Спецификация пересылки и останова может быть задана с помощью добавления тэговой битовой отметки (битов пересылки и стоповых битов) к каждой команде в задаче. Возможно также другое введение битовых отметок. Например, с каждой статической командой может быть ассоциирована таблица тэговых битов. Аппаратные средства выбирают команды из текста программы и соответствующие битовые отметки из таблицы, объединяют эту пару в новую команду и помещают в кэш-память команд. Освобождение регистра может быть задано добавлением специальной команды освобождения базовой системе команд.

Мультискалярная программа может быть сгенерирована из существующей двоичной, путем добавления описателей задачи и битовых отметок. Эта мультискалярная информация может быть размещена внутри кода программы, а также до или после кода.

Разделение исполнительного кода и описателей позволяет упростить процедуру перенесения программы на другие аппаратные средства.

Мультискалярные аппаратные средства.

На мультискалярные аппаратные средства возлагаются функции обхода ГУЗ, назначения задач на ПЭ и выполнения этих задач при сохранении последовательной семантики программы. Работа по определению порядка назначения задач на выполнение возлагается на программу-планировщик. По адресу описателя задачи программа-планировщик выбирает описатель задачи и назначает задачу на ПЭ, выдает адрес первой команды, устанавливает маски создания и накопления для задачи. Планировщик, используя статическую или динамическую схему предсказания, на основе информации из описателя задачи, предсказывает задачу-преемницу. Каждый ПЭ независимо выбирает и выполняет команды задачи до тех пор, пока не сталкивается с командой останова, идентифицирующей завершение задачи.

Основной целью распределения задач по ПЭ мультискалярного процессора является создание возможности выполнения нескольких команд в одном такте. Потери производительности мультискалярного процессора возможны из-за наличия у ПЭ тактов бесполезных вычислений, тактов ожидания и свободных тактов.

Бесполезные такты вычисления представляют работу, которая в последующем будет отменена в виду использования неправильных значений данных или неправильного предсказания. Такты ожидания связаны с ожиданием получения значения, созданного командой в задаче-предшественнице, или значения, созданного командой в той же самой задаче (например, в случае операции с большим временем выполнения или в виду неудачного обращения в кэш), а также с ожиданием при выполнении действий

по перепланированию задач. Неактивные такты составляют время, в течение которого ПЭ не имеет назначенной задачи. Основной причиной их возникновения является несбалансированность загрузки ПЭ.

Для уменьшения потерь, связанных с бесполезными тактами, следует снижать вероятность отмены результатов предыдущих вычислений, обеспечивая синхронизацию ПЭ по данным, а также устанавливая факт необходимости отмены (если он имеет место быть) как можно раньше.

Преимущества мультискалярной архитектуры.

Мультискалярный процессор имеет некоторые свойства, выгодно отличающие его от традиционных суперскалярных микропроцессоров.

При стандартном подходе точность предсказания ветвлений ограничивает степень параллелизма. Если средняя вероятность правильного предсказания перехода - 0,9, то вероятность правильного предсказания на пять ветвлений вперед только 0,6.

Мультискалярный процессор имеет большую глубину предсказания при обеспечении высокой вероятности выбора правильного направления вычислений. Это свойство обусловлено избирательностью предсказания ветвей. Мультискалярный процессор разбивает последовательный поток команд на задачи. Хотя задачи могут содержать внутренние ветви, планировщик должен предсказывать только ветви, которые отделяют задачи. Ветви, содержащиеся внутри задачи, не предсказываются (если они не предсказаны отдельно внутри ПЭ).

Для суперскалярных процессоров наличие широкого окна выполнения приводит к увеличению числа отложенных команд и усложняет контроль результатов выполнения всех команд в этом окне.

В мультискалярной реализации окно может быть очень широким, однако в любой момент времени только несколько команд должны быть рассмотрены на предмет выдачи результатов (только одна для каждого ПЭ). Границы окна отложенных команд могут быть идентифицированы первой и последней командами в очереди на исполнение.

Для одновременной выдачи n результатов в процессоре должна использоваться логика со сложностью n^2 , чтобы выполнить перекрестную проверку зависимостей среди команд. В суперскалярном процессоре, это ограничивает пропускную способность логики выдачи. В мультискалярном процессоре каждый ПЭ выдает команды независимо, т.е. используется сложность логики порядка n .

Прежде чем переупорядочить доступ к памяти, необходимо идентифицировать и вычислить все адреса загрузки и записи значений.

В суперскалярной реализации команды загрузки и записи упорядочиваются (или сохраняются в первоначальной последовательности) и помещаются в буфер вместе с адресом доступа к памяти. При выполнении команды загрузки проверяется буфер, чтобы гарантировать, что не отложена никакая более

ранняя команда записи по тому же самому или еще не определенному адресу. При выборке значения из памяти буфер проверяется, чтобы гарантировать, что не отложена никакая более ранняя команда загрузки или записи по тому же самому или еще не определенному адресу. В мультискалярной реализации команды загрузки и записи могут быть выполнены независимо, без знания последовательности выполнения команд загрузки и записи в задачах преемнице или предшественнице.

В суперскалярном процессоре возможна генерация достаточно широкого окна выполнения с большой глубиной предсказания ветвлений. Кроме того, возможно генерировать очень гибкий план выполнения команд. Например, загрузка в вызываемой функции может выполняться параллельно с запоминанием в вызывающей функции. Однако суперскалярный процессор не имеет представления о ГУЗ программы. Поэтому и возникает необходимость предсказания каждого перехода, что, в конечном счете, приводит к снижению точности предсказания и производительности.

Мультискалярный процессор во многом похож на многопроцессорную систему с общей памятью и очень низким уровнем непроизводительных затрат на планирование. Главное их отличие заключается в том, что многопроцессорная система требует, чтобы компилятор делил программу на задачи, где все соотношения зависимостей между задачами известны (предусмотрены программистом путем использования операторов синхронизации и межпроцессорных коммуникаций), а мультискалярный процессор не требует никакого априорного знания относительно связей команд по управлению и данным.

Мультискалярная архитектура объединяет принципы низко- и высокоуровневого распараллеливания, методы анализа статической и динамической структур программы, благодаря чему позволяет добиться более высоких значений эффективности использования вычислительных ресурсов процессора, чем другие типы архитектур. Фактически в мультискалярных процессорах реализован симбиоз автоматически распараллеливающего компилятора, дающего указания аппаратуре процессора в виде отметок команд и специальных команд, и аппаратных средств, воспринимающих эти указания.

Изложенный выше подход не является единственно возможным при реализации этой плодотворной идеи - привлечения компилятора к распределению заданий по ПЭ и балансировке загрузки процессоров в многопроцессорных системах.

Лекция 8.

Организация интерфейса МП устройств с внешними устройствами и памятью. Встроенные микропроцессорные системы управления.

Появление в 1971 году первого микропроцессора ознаменовало собой начало новой эры не только в вычислительной технике, но и в области *систем встроенного управления оборудованием* — эры высокопроизводительных и

надежных цифровых, микропроцессорных систем управления, интегрированных в рабочую машину, механизм, прибор, изделие — станок, робот, стиральную машину, принтер, плоттер, радиотелефон, автомобиль и т.д.

Компьютерная и связанная с ней микропроцессорная техника развиваются столь стремительно, что производительность процессоров удваивается каждые 8-10 месяцев при значительном росте степени интеграции, снижении габаритов, энергопотребления и цены.

Революция в управляющей электронике сопровождается революцией и в силовой электронике, что позволяет создавать *интегрально-гибридные интеллектуальные электронные модули*, а также конструктивно интегрировать в одном изделии рабочий орган механизма, силовой преобразователь, устройство управления, источник питания и датчики.

Вместо традиционного термина «электромеханика» постепенно начинают применять термин «мехатроника», означающий конструктивное объединение механики и электроники на принципиально новом уровне, когда рост управляющих вычислительных ресурсов при использовании *больших интегральных схем* (БИС) сопровождается значительным повышением надежности оборудования, главным образом, за счет резкого уменьшения монтажных соединений и расширением функциональных возможностей изделия в сторону автоматической адаптации к условиям эксплуатации и интерактивного обучения в процессе диалога с оператором.

Встроенной системой управления будем называть систему управления, конструктивно интегрированную в оборудование. Так, система управления, встроенная в статический преобразователь частоты для асинхронных двигателей, может представлять собой одно- или многоплатную микроЭВМ с необходимым набором интерфейсов для обеспечения как непосредственного управления инвертором и приводом в целом, так и взаимодействия с человеком-оператором, включая, например пульт оперативного управления с дисплеем и клавиатурой. Обязательным компонентом такой системы является интерфейс с системой управления более высокого уровня (промышленным программируемым контроллером или компьютером в промышленном исполнении), что позволяет решать задачи комплексной автоматизации с использованием заданного числа единиц технологического оборудования, объединяя это оборудование в единую *распределенную систему управления.*

В зависимости от сложности решаемой задачи встроенная система управления может быть *однопроцессорной* или *многопроцессорной (мультипроцессорной)*, *одноуровневой* или *многоуровневой.* В последнем случае на *нижнем уровне управления* решаются задачи *непосредственного управления* отдельными компонентами оборудования (например отдельными осями привода, узлами автомобиля), а на следующем, более высоком, - задачи совместного управления в реальном времени, связи с оператором, системой верхнего уровня и т.д. Типичными примерами являются станки с числовым программным управлением, роботы-манипуляторы, изделия автомобильной и авиационной электроники, системы связи.

Для многоуровневых систем управления в настоящее время четко прослеживается тенденция оформления систем управления *среднего уровня* в самостоятельном конструктиве для монтажа в стойки или в панели управления. Возможен также вариант настоольного или настенного монтажа. Такие устройства управления получили название *индустриальных рабочих станций, индустриальных компьютеров, панельных компьютеров, промышленных программируемых контроллеров*. При этом контроллеры нижнего уровня управления могут подключаться к системе среднего уровня управления несколькими способами: через стандартные последовательные или параллельные интерфейсы; путем установки на системную шину в качестве специализированных *устройств сопряжения с объектом (модулей УСО)*.

Индустриальные компьютеры имеют, как правило, развитый набор стандартных устройств сопряжения с объектом (УСО), делающий их универсальными средствами управления для широкого круга применений. Тем не менее на нижнем уровне управления двигателями, статическими преобразователями и др. требуется в большинстве случаев разработка *встроенных специализированных контроллеров*, причем эту разработку могут выполнить только специалисты в данной предметной области, хорошо разбирающиеся в особенностях объекта управления, способные предложить и реализовать весь комплекс необходимых алгоритмов оптимального управления объектом.

Главное отличие *промышленного контроллера* от *промышленного компьютера* состоит в адаптации языка программирования под конкретную область применения. Так, большинство промышленных контроллеров, выпускаемых такими известными фирмами как Allen Bradley, Siemens, Fanuc, АВВ и др. являются *программируемыми логическими контроллерами (PLC)* и имеют встроенные интерпретаторы с языка релейной автоматики или языка Булевой алгебры, что позволяет неспециалисту программирование системы управления. Таким образом, одноуровневые системы встроенного управления выполняются преимущественно на *базе однокристальных микроЭВМ - микроконтроллеров*, а двух- и более -уровневые предполагают использование на среднем уровне управления многоплатных микроЭВМ типа промышленного программируемого контроллера или компьютера в промышленном исполнении.

Встраиваемое управляющее устройство является *микропроцессорной системой управления*, в состав которой кроме центрального процессора на базе однокристального микропроцессора и микроконтроллера входят необходимые дополнительные элементы памяти и периферийные интерфейсные БИС для организации сопряжения с датчиками, объектом управления и системой управления более высокого уровня.

С точки зрения производителей микропроцессорной техники все задачи, решаемые системами встроенного управления делятся на два больших класса: *управление событиями в реальном времени и управление потоками данных*. Каждый класс задач предъявляет свои специфические требования к микропроцессору или микроконтроллеру, что отражается прежде всего в наборе функций, реализуемых на кристалле, а также в системе команд.

К первому классу относятся задачи, требующие быстрой реакции микропроцессорной системы на изменение внешних условий (на срабатывание технологических датчиков, изменение параметров и т.д.). Как правило, системы управления приводами, энергетическими установками, роботами, а также системы распределенной автоматизации относятся к системам первого класса. Эти задачи требуют применения микроконтроллеров с большим объемом интегрированной на кристалл периферии, включая реализацию на кристалле памяти программ, памяти данных и устройств ввода-вывода, что сокращает аппаратные затраты и удешевляет изделие со встроенной системой управления. Чаще всего в системах управления этого класса для реализации алгоритма управления требуется память относительно небольшого объема (до 32 Кбайт).

Ко второму классу задач относятся задачи, требующие быстрой обработки значительных объемов информации, например в микропроцессорных системах поддержки компьютерных сетей, в системах управления летательными аппаратами, подвижным составом, в системах обработки видеоизображений, когда встроенный процессор должен выполнять множество различных вычислительных операций, в том числе операций с плавающей запятой. Как правило, для решения таких задач требуется уже высокопроизводительный 32- или 64-разрядный процессор.

В соответствии с упомянутыми выше классами задач, продукция фирмы Intel для встроенных применений может быть разделена на следующие группы:

- 8-битовые микроконтроллеры первого поколения (семейство MCSR-48).
- Современные 8-разрядные микроконтроллеры (MCSR-51, MCSR-51, MCSR-251).
- Современные 16-разрядные микроконтроллеры для управления в реальном времени (MCSR-96, MCSR-196, MCSR.-296).
- Встраиваемые 16-разрядные и 32-разрядные микропроцессоры PC-подобной архитектуры (80C186, 386EX и др.).
- Высокопроизводительные микропроцессоры, построенные по RISC-технологии (1960).

Первые три группы изделий ориентированы на управление событиями в реальном времени. Две последние группы предназначены преимущественно для управления процессами. Из изделий, предназначенных для управления событиями, группа 16-разрядных микроконтроллеров MCSR-96 является наиболее производительной и имеет богатейший выбор интегрированных на кристалл специализированных периферийных устройств. Основной базой для построения встроенных систем управления нижнего уровня являются именно *однокристалльные микроЭВМ и микроконтроллеры*, могут применяться также и законченные одноплатные системы управления на их основе, выпускаемые рядом фирм в качестве *контроллеров-прототипов*. Долгое время в нашей стране были доступны в основном только освоенные отечественной промышленностью младшие модели 8-разрядных микропроцессоров и микроконтроллеров фирмы Intel, совместимые с изделиями MCS-80, MCS-48,

MCS-51, а также 16-разрядные процессоры собственной разработки, система команд которые совместимы с процессорами фирмы DEC (1816BM1, 1816BM2). В настоящее время ситуация резко изменилась, и вся самая передовая продукция ведущих фирм мира стала доступна отечественному разработчику и производителю.

Сегодня разработчики встроенных систем управления стоят перед непростым выбором: какое изделие и какой фирмы использовать в проекте.

Разработка мощной микропроцессорной системы управления является сложным делом. Для этого необходим штат высококвалифицированных инженеров-схемотехников и программистов, а также соответствующее оборудование, а именно комплект аппаратных и программных средств разработки (кстати, для каждого набора микропроцессорных БИС-свой). Современные микропроцессоры и микроконтроллеры имеют высокие тактовые частоты и требуют исключительной тщательности как при проектировании печатной платы, включая разводку, так и при ее изготовлении.

На сегодняшний день общемировая потребность в микроконтроллерах составляет более 600 млн. штук в год и быстро возрастает. Это говорит о массовом освоении промышленностью изделий с высокопроизводительными встроенными системами управления в самых разнообразных отраслях техники. Эти системы находят применение в автомобильной промышленности, станко- и роботостроении, в самолетной индустрии, в производстве периферийных устройств компьютеров (приводы дисковых накопителей, скоростные принтеры, плоттеры и т.д.), в бытовой технике (стиральные машины, видеомагнитофоны и т.д.) и, разумеется, в энергетике и электромеханике, где 16-разрядные устройства управления появились даже в относительно несложных приводах бытовых кондиционеров. Особенно широкие возможности по применению этих контроллеров открываются перед разработчиками систем автоматизированного привода, стабилизированных источников питания, промышленных контроллеров и других средств автоматизации производственных процессов.

Лекция 9.

Однокристалльные микро-ЭВМ.

В микропроцессорной технике выделился самостоятельный класс больших интегральных схем (БИС) - однокристалльные микроЭВМ (микроконтроллеры), которые предназначены для "интеллектуализации" оборудования различного технологического назначения. Архитектура однокристалльных микроЭВМ - результат эволюции архитектуры микропроцессоров и микропроцессорных систем, обусловленный стремлением существенно снизить их аппаратные затраты и стоимость.

Однокристалльные микроЭВМ представляют собой приборы, конструктивно выполненные в виде одной БИС и включающие в себя все устройства, необходимые для реализации цифровой системы управления минимальной конфигурации: процессор, запоминающее устройства

данных, запоминающее устройство команд, внутренний генератор тактовых сигналов, а также программируемые схемы для связи с внешней средой.

Использование однокристалльных микроЭВМ в системах управления обеспечивает достижение исключительно высоких показателей эффективности при столь низкой стоимости, что им, видимо, нет в ближайшем времени альтернативной элементной базы для построения управляющих и/или регулирующих систем.

Отечественная микроэлектронная промышленность освоила широкомасштабный выпуск однокристалльных микроЭВМ, к числу которых можно отнести 4-битные микроконтроллеры серий 1814, 1820, 1829 и 1013; 8-битные микроЭВМ серии 1816; микроконтроллеры сигнальные (аналоговые микропроцессоры) серии 1813. В данном курсе в качестве базовых выбраны однокристалльные микроЭВМ (ОМЭВМ) серии 1816. В настоящее время в состав этой серии входят два семейства ОМЭВМ МК48 и МК51.

Общие сведения об однокристалльных микроЭВМ семейства МК48.

Семейство МК48 включает ряд моделей ОМЭВМ, функциональный состав и технические характеристики которых отражают как различные в идеологическом подходе к применению ОМЭВМ, так и прогрессе технологии СБИС. Все модели, входящие в семейство МК48, являются полностью совместимыми по системе команд, назначению и разводке выводов, совокупности основных функциональных устройств из базового набора семейств.

Первое поколение отечественных ОМЭВМ семейства МК48 - БИС КР1816ВЕ48 и КР1816ВЕ35 являются функционально-конструктивными аналогами БИС соответственно 8748 и 8035 фирмы Intel (США), выполнены по n-канальной МОП-технологии, что обусловило следующие ограничения: уровень интеграции до 18 тыс. транзисторов на кристалле, частота следования тактовых сигналов - 6.0 МГц, объем внутренней памяти ОЗУ - 64 байта, ПЗУ - 1 кбайт и минимальное время цикла - 2,5 мкс.

Второе поколение - БИС КР1816ВЕ49, КР1816ВЕ39 (аналоги БИС 8049 и 8039 фирмы Intel) выполнено по n-канальной МОП-технологии с пропорциональным масштабированием, что позволило повысить уровень интеграции до 36 тыс. транзисторов на кристалле, частоту следования тактовых сигналов до 11МГц, увеличить объем ОЗУ до 128 байт, ПЗУ до 2 кбайт и снизить минимальное время цикла до 1,36 мкс.

Третье поколение семейства МК48 - БИС ОМЭВМ серии КР1830: КР1830ВЕ48, КР1830ВЕ35 (аналоги БИС 80С48, 80С35 фирмы Intel) выполнено по КМОП-технологии, что позволило на порядок уменьшить ток потребления по сравнению с БИС КР1816 ВЕ48, КР1816ВЕ35 при сохранении остальных параметров.

ОМЭВМ КР1816ВЕ48, КР1816ВЕ35, КР1830ВЕ48 и КР1830ВЕ35 полностью идентичны в части структурной реализации. При этом в БИС

КМ1816ВЕ48 программная память размещается во внутреннем ППЗУ с ультрафиолетовым (УФ) стиранием, а в БИС КР1830ВЕ48 - во внутреннем ПЗУ масочного типа. Таким образом, оперативность программирования ППЗУ позволяет использовать ОМЭВМ КР1816ВЕ48 при создании контроллеров единичных экземпляров или мелкосерийных изделий. Потребители БИС КР1830ВЕ48 лишены такой возможности, так как программирование ПЗУ осуществляется в процессе изготовления БИС по данным "прошивки" заказа потребителя.

В микросхемах КР1816ВЕ35 и КР1830ВЕ35 в отличие от БИС КМ1816ВЕ48, КР1830ВЕ48 память программ реализуется только за счет подключения внешней памяти любого типа (ОЗУ, ППЗУ, ПЗУ) общим объемом до 4 кбайт. Эта особенность позволяет использовать их в качестве отладочного варианта, когда память программ реализуется в ОЗУ, что позволяет легко модифицировать отлаживание программ.

ОМЭВМ КР1816ВЕ49 и КР1816ВЕ39 имеют одну и ту же структуру, одинаковые схемотехнические решения и технические характеристики, за исключением памяти программ: ОМЭВМ КР1816ВЕ49 имеет внутреннюю память программы объемом 2 кбайт, выполненную в виде масочного ПЗУ, а ОМЭВМ КР1816ВЕ39 может использоваться только с внешней ЗУ программ.

Реализация программной памяти КР1816ВЕ49 в виде ПЗУ обуславливает целесообразность применения этих ОМЭВМ только для изделий средне- и крупносерийного производства, что обеспечивает в этом случае низкую стоимость ОМЭВМ. В качестве отладочной модели, а также при разработке единичных экземпляров изделий целесообразно использовать ОМЭВМ КР1816ВЕ39 с внешней памятью программ.

В общем виде основные отличительные особенности ОМЭВМ семейства МК48 представлены в таблице 1.

Таблица 1

Микро- схемы	Ана лог	Объ ем вну тр. пам яти	Тип памяти програ мм	Объ ем Пам яти Данн ых	Макс. частота след.тактов ых сигналов	еб я
				байт	МГц	
КР1826В Е35	803 5	НЕ Т	ВНЕШ Н.	64 64	6.0 6.0	0
КР1816В Е48	874 8	1К НЕ	УФПП ЗУ	128 128	11.0 11.0	0

КР1816В Е39	803 9	Т 1К	ВНЕШ Н.	64 64	64 64	0 0
КР1816В Е49	804 9	НЕ Т	ПЗУ ВНЕШ			
КР1830В Е35	80С 35	1К	Н. ПЗУ			
КР1830В Е48	80С 48					

В ОМЭВМ предусмотрена возможность расширения памяти программ до 4 кбайт, памяти данных до 384 байт и увеличения числа линий ввода вывода за счет подключения внешних кристаллов памяти программ, ОЗУ и БИС интерфейсов.

Однокристалльные микро-ЭВМ семейства МК51.

8-разрядные высокопроизводительные однокристалльные микроЭВМ семейства МК51 выполнены по высококачественной n-МОП-технологии (серия 1816) и КМОП-технологии (серия 1830).

Использование ОМЭВМ семейства МК51 по сравнению с МК48 обеспечивает увеличение объема памяти программ и памяти данных. Новые возможности ввода-вывода и периферийных устройств расширяют диапазон применения и снижают общие затраты системы. В зависимости от условий использования, быстродействие системы увеличивается минимум в два с половиной раза и максимум на порядок.

Семейство МК51 включает пять модификаций ОМЭВМ (имеющих идентичные основные характеристики), различающиеся реализацией памяти программ и мощностью энергопотребления.

ОМЭВМ КР1816ВЕ51 и КР1830ВЕ57 содержат масочно-программируемое в процессе изготовления кристалла ПЗУ памяти программ емкостью 4 кбайт и рассчитаны на применение в массовой продукции. За счет использования внешних микросхем памяти общий объем памяти программ может быть расширен до 64 кбайт.

ОМЭВМ КМ1816ВЕ751 содержит ППЗУ емкостью 4 кбайт с УФ стиранием и удобна на этапе разработки системы при отладке программ, а также при производстве небольшими партиями или при создании систем, требующих в процессе эксплуатации периодической подстройки. За счет использования внешних микросхем памяти общий объем памяти программ может быть расширен до 64 кбайт.

ОМЭВМ КР1816ВЕ31 и КР1830ВЕ31 не содержат встроенной памяти программ, однако могут использовать до 64 кбайт внешней постоянной или перепрограммируемой памяти программ и эффективно использоваться в системах, требующих существенно большего по объему (более чем 4 кбайт на

кристалле) ПЗУ памяти программ.

Каждая из перечисленных выше микросхем является соответственно аналогом БИС 8051, 80С51, 8751, 8031, 80С31 семейства MCS-51 фирмы Intel (США).

Сравнительные данные микросхем МК51 приведены в таблице 2.

Таблица 2

МИКРО-СХЕМЫ	АНАЛОГ	ОБЪЕМ ВНУТР. ПАМЯТИ	ТИП ПАМЯТИ ПРОГРАММ	ОБЪЕМ ПАМЯТИ ДАННЫХ	МАКС. ЧАСТОТА СЛЕД. ТАКТОВ. СИГНАЛОВ	ТОК ПОТРЕБЛЕНИЯ
		БАЙТ		БАЙТ	МГц	мА
КР1826ВЕ31	8031АН	НЕТ	ВНЕШН.	128	12.0	150.0
КР1816ВЕ51	8051АН	4К	ПЗУ	128	12.0	150.0
КР1816ВЕ751	8751Н	4К	ППЗУ	128	12.0	220.0
КР1830ВЕ31	80С31ВН	НЕТ	ВНЕШН.	128	12.0	18.0
КР1830ВЕ51	80С51ВН	4К	ПЗУ	128	12.0	18.0

Каждая ОМЭВМ рассматриваемого семейства содержит встроенное ОЗУ памяти данных емкостью 128 байт с возможностью расширения общего объема оперативной памяти данных до 64 Кбайт за счет использования внешних микросхем ЗУПВ.

Общий объем памяти ОМЭВМ семейства МК51 может достигать 128 Кбайт: 64 Кбайт памяти программ и 64 Кбайт памяти данных.

ОМЭВМ МК51 выполнены на основе высокоуровневой n-МОП-технологии и выпускаются в корпусе типа DIP, имеющем 40 внешних выводов. ОМЭВМ данного семейства содержат следующие основные узлы:

1. центральный 8-разрядный процессор;
2. память программ объемом 4 Кбайт;
3. память данных объемом 128 байт;
4. четыре 8-разрядных программируемых канала ввода-вывода;
5. два 16-битовых многорежимных таймера/счетчика;
6. систему прерываний с пятью векторами и двумя уровнями;
7. последовательный интерфейс;
8. тактовый генератор.

Система команд ОМЭВМ содержит 111 базовых команд с форматом 1,2 или 3 байта.

ОМЭВМ имеет: 32 РОН; 128 определяемых пользователем программно-управляемых флагов; набор регистров специальных функций.

РОН и определяемые пользователем программно-управляемые флаги расположены в адресном пространстве внутреннего ОЗУ данных.

Структурная схема ОМЭВМ представлена на рис.10. Основу

структурной схемы МК51 составляет внутренняя двунаправленная 8-битовая шина, которая связывает между собой все основные узлы и устройства: арифметико-логическое устройство (АЛУ), резидентную память, блок регистров специальных функций, устройство управления и порты ввода-вывода.

Устройство управления МК51 на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам кварцевого резонатора или шести состояниям первичного управляющего автомата (S1-S6). Каждое состояние управляющего автомата содержит две фазы (P1, P2) сигналов резонатора. В фазе P1, как правило, выполняется операция в АЛУ, а в фазе P2 осуществляется межрегистровая передача. Весь машинный цикл состоит из 12 фаз, начиная с фазы S1P1 и кончая фазой S6P2.

Арифметико-логическое устройство. 8-битное АЛУ может выполнять арифметические операции сложения, вычитания, умножения и деления, логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. В АЛУ имеются программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции и схема формирования признаков.

Важной особенностью АЛУ является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Вследствие этого, МК51 иногда называют "булевым процессором".

Таким образом, АЛУ может оперировать четырьмя типами объектов: булевыми (1 бит), цифровыми (4 бита), байтовыми (8 бит), адресными (16 бит).

Резидентная память. Резидентная память МК51 включает в себя память программ (РПП) и память данных (РПД). РПП и РПД физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции.

РПП имеет емкость 4Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации и т.п. Она имеет 16-битную шину адреса, по которой обеспечивается доступ из счетчика команд (PC) или из регистра указателя данных (DPER). Последний выполняет функции базового регистра при косвенных переходах.

РПД (ОЗУ) предназначена для хранения переменных в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к адресному пространству РПД примыкают адреса регистров специальных функций (SFR).

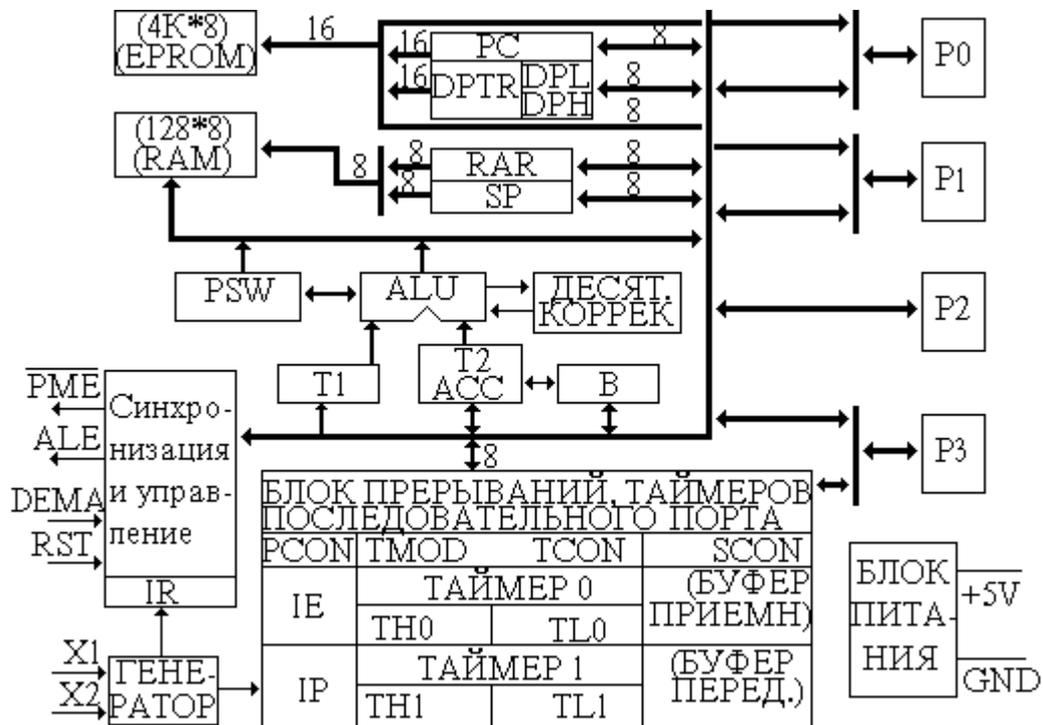


РИС.10. СТРУКТУРНАЯ СХЕМА ОМЭВМ

На рис. 10. приняты следующие обозначения для управляющих сигналов:

/PME- разрешение внешней памяти программ, сигнал выдается только при обращении к внешней памяти программ;

ALE - строб адреса внешней памяти, сигнал используется для приема и фиксации адреса внешней памяти на внешнем регистре;

DEMA - блокирование работы с внутренней памятью, отключение резидентной памяти программ;

RST - сигнал общего сброса, служит также выводом резервного питания ОЗУ от внешнего источника;

X1, X2 - выводы для подключения кварцевого резонатора.

РПП и РПД могут быть расширены до 64 Кбайт путем подключения внешних БИС.

Регистры аккумулятора и PSW (регистр признаков операций). Аккумулятор является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных.

При выполнении многих команд в АЛУ формируется ряд признаков операций (флагов), которые фиксируются в регистре PSW.

Регистры указатели. 8-битный указатель стека (SP) может адресовать любую область РПД. Его содержимое инкрементируется прежде, чем данные будут запомнены в стеке в ходе выполнения команды PUSH и CALL. Содержимое SP декрементируется после выполнения команд RET и POP. Подобный способ адресации элементов стека называют прединкрементным/постинкрементным.

Двухбайтовый регистр указатель данных (DPTR) обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней памяти. DPTR может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

Буфер последовательного порта. Регистр с символическим именем SBUF представляет собой два независимых регистра - буфер приемника и буфер передатчика. Загрузка байта в SBUF автоматически вызывает начало процесса передачи через последовательный порт.

Когда байт считывается из SBUF, это значит, что его источником является приемник последовательного порта.

Блок таймеров/счетчиков. В составе средств МК51 имеются регистровые пары с символическими именами TH0, TL0 и TH1, TL1, на основе которых функционируют два независимых программно-управляемых 16-битных таймера/счетчика событий. Таймеры/счетчики (Т/С) предназначены для подсчета внешних событий, для получения программно-управляемых временных задержек и выполнения время-задающих функций ОМЭВМ.

В состав блока Т/С входят: два 16-разрядных регистра Т/С0 и Т/С1; 8-разрядный регистр режимов Т/С (ТМ0D); 8-разрядный регистр управления (ТС0N); схема инкремента; схема фиксации INT0, INT1, T0, T1; схема управления флагами.

16-разрядные регистры Т/С выполняют функцию хранения содержимого счета. Код начального счета заносится в указанные регистры программно. В процессе счета содержимое регистров инкрементируется. Признаком окончания счета является переполнение регистров.

Регистр ТМ0D хранит код, определяющий:

- один из 4-х возможных режимов работы каждого Т/С;
- работу в качестве таймеров или счетчиков;
- управление Т/С от внешнего вывода.

Регистр ТС0N предназначен для приема и хранения кода управляющего слова

При работе в качестве таймера содержимое Т/С инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов резонатора. При работе в качестве счетчика содержимое Т/С инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (Т0, Т1) вывод МК51. Опрос значения внешнего входного сигнала выполняется в момент времени S5P2 каждого машинного цикла. Содержимое счетчика будет увеличено на 1 в том случае, если в предыдущем цикле был считан входной сигнал высокого уровня (1), а в следующем - сигнал низкого уровня (0). Новое (инкрементированное) значение счетчика будет сформировано в момент S3P1 в цикле, следующем за тем, в котором был обнаружен переход сигнала из 1 в 0. Так как на распознавание перехода требуется два машинных цикла, то максимальная

частота подсчета входных сигналов равна $1/24$ частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входного считываемого сигнала он должен удерживать значение 1 как минимум в течении одного машинного цикла МК51.

Режим работы 0,1 и 2 для обоих Т/С одинаковы. Режим 3 для Т/С0 и Т/С1 различны.

Рассмотрим кратко работу Т/С во всех режимах.

В режиме 0 таймерный регистр имеет разрядность 13 бит. При переходе из состояния "все единицы" в состояние "все нули" устанавливается флаг прерывания от таймера TF. Входной синхросигнал таймера разрешен (поступает на вход Т/С), когда управляющий бит TR установлен в 1 и либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания INT поступает уровень 1.

Отметим попутно, что установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала, подаваемого на вход запроса прерывания.

Работа любого Т/С в режиме 1 такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.

В режиме 2 работа организована таким образом, что переполнение 8-битного счетчика TL приводит не только к установке флага TF, но и автоматически перезагружает в TL содержимое старшего байта TH таймерного регистра, которое предварительно было задано программным путем.

Т/С1 в режиме 3 заблокирован и просто сохраняет значение кода в регистре Т/С.

Т/С0 в режиме 3 представляет собой два независимых устройства на основе 8-ми разрядных регистров TL0 и TH0. Устройство на основе TL0 может работать в режиме таймера и в режиме счетчика, за ним сохраняются все биты управления Т/С0, оно реагирует на воздействия по входам T0, INT0. При переполнении TL0 устанавливается флаг TF0. Устройство на основе регистра TH0 может работать только в режиме таймера. Оно использует бит включения TR1, при переполнении TH0 выставляет флаг TF1.

Порты. Порты P0, P1, P2, P3 являются двунаправленными портами ввода-вывода и предназначены для обмена информацией ОМЭВМ с внешними устройствами, образуя 32 линии ввода-вывода. Каждый из портов содержит фиксатор-защелку, который представляет собой 8-ми разрядный регистр, имеющий байтовую и битовую адресацию для установки (сброса) разрядов с помощью программного обеспечения.

Помимо работы в качестве обычных портов ввода-вывода линии портов P0-P3 могут выполнять ряд дополнительных функций.

Через порт P0:

- выводится младший байт адреса A0-A7 при работе с внешней памятью

программ и внешней памятью данных;

- выдается и принимается байт данных при работе с внешней памятью (при этом обмен байтом данных и вывод младшего байта адреса внешней памяти мультиплексируются по времени);

- задаются данные при программировании внутреннего ППЗУ и читается содержимое внутренней памяти программ.

Через порт P1:

- задается младший байт адреса при программировании внутреннего ППЗУ и при чтении внутренней памяти программ.

Через порт P2:

- выводится старший байт адреса A8-A15 при работе с ВПП и ВПД;

- задается старший байт адреса (A8-A14) при программировании внутреннего ППЗУ и при чтении внутренней памяти программ.

Каждая линия порта P3 имеет индивидуальную альтернативную функцию. Альтернативная функция любой из линий порта P3 реализуется только в том случае, если в соответствующем этой линии разряде фиксатора защелки содержится "1". В противном случае на линии порта будет присутствовать "0".

Организация обращения к ВПД. В ОМЭВМ предусмотрена возможность расширения памяти данных путем подключения внешних устройств емкостью до 64 Кбайт. Команды ОМЭВМ могут формировать 8-разрядный адрес ВПД, который выдается через порт P0, либо 16-разрядный адрес, младший байт которого выдается через порт P0, а старший - через порт P2. Байт адреса, выдаваемый через порт P0, должен быть зафиксирован во внешнем регистре по спаду сигнала ALE, т.к. в дальнейшем линии порта P0 используются как шина данных, через которую байт данных принимается из памяти при чтении или выдается в память данных при записи. При этом чтение строится сигналом RD, а запись - сигналом WR. Схема подключения внешней памяти данных к ОМЭВМ показана на рис.11.

Пространства внутренней и внешней памяти данных не пересекаются, т.к. доступ к ним осуществляется с помощью разных команд. Т.о. в системе могут одновременно присутствовать внутренняя память данных с адресом 00H-FFH и внешняя память данных с адресом 0000H-FFFFH. Обращение к ячейкам ВПД осуществляется только с использованием косвенной адресации по регистрам R0 и R1 активного банка регистров внутреннего ОЗУ или по регистру специальных функций DPTR. Соответственно в первом случае будет формироваться 8-разрядный адрес ВПД.

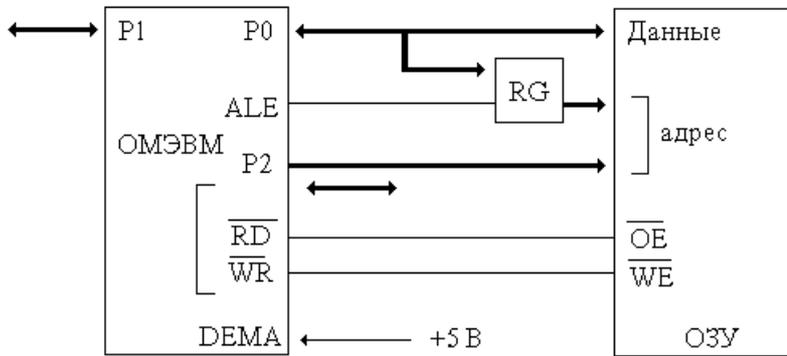


Рис.11. Схема подключения внешней памяти данных к ОМЭВМ

На рис.12а, 12б, соответственно, приведены диаграммы циклов чтения и записи при работе ОМЭВМ с внешней памятью данных.

DPL, DPH - соответственно младший и старший байты регистра указателя данных DPTR, который используется в качестве регистра косвенного адреса. R0 - регистр R 0 текущего банка, который используется в качестве регистра косвенного адреса.

P2SFR - защелка порта P2.

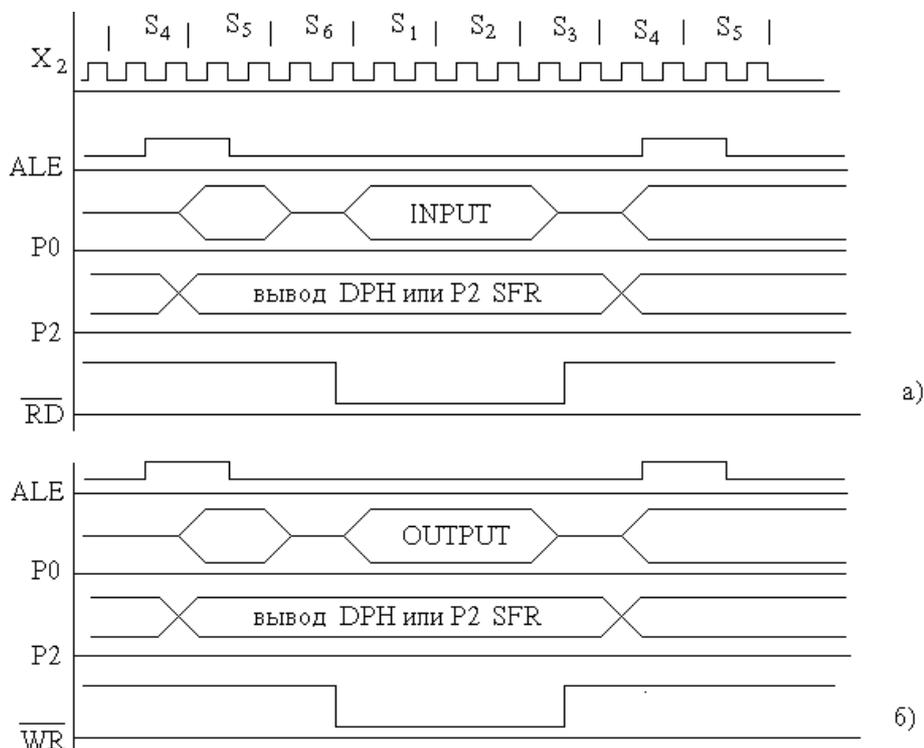


Рис.12. Диаграммы циклов чтения и записи при работе ОМЭВМ с внешней памятью данных

Память программ. Память программ предназначена для хранения кодов программ и имеет отдельное от памяти данных адресное пространство объемом до 64 Кбайт, причем для БИС 1816BE51, 1816BE75 и 1830BE51 часть памяти программ с адресами 0000H-0FFFH расположена на кристалле ОМЭВМ. Если на вывод DEMA подано напряжение U_{cc} , то обращение к ВПП происходит автоматически при выработке счетчиком команд адреса,

превышающего 0FFFH.

Если на вывод DEMA подан "0", внутренняя память программ отключается и начиная с адреса 0000H все обращения выполняются только к внешней памяти программ.

Чтение из внешней памяти программ стробируется сигналом ОМЭВМ XOR(PME). При работе с внутренней памятью программ сигнал XOR(PME) не формируется.

При обращении к ВПП всегда формируется 16-разрядный адрес, младший байт которого выдается через порт P0, а старший - через порт P2. При этом байт адреса, выдаваемый через порт P0, должен быть зафиксирован во внешнем регистре по спаду сигнала ALE, т.к. в дальнейшем линии порта P0 используются в качестве шины данных, по которой байт из внешней памяти программ вводится в ОМЭВМ.

Функциональная схема включения ОМЭВМ МК51 с ВПП показана на рис.13.

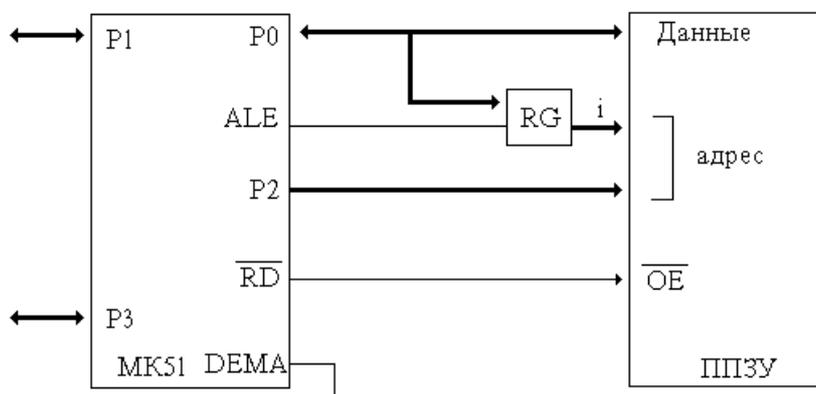


Рис.13. Функциональная схема включения ОМЭВМ МК51 с ВПП

Порт P0 работает как мультиплексированная шина адрес/данные: выдает младший байт счетчика команд, а затем переходит в высокоимпедансное состояние и ожидает прихода байта из ППЗУ программ. Когда младший байт адреса находится на выходах порта P0, сигнал ALE защелкивает его в адресном режиме RG. Старший байт адреса находится на выходах порта P2 в течение всего времени обращения к ППЗУ.

Сигнал XOR(PME) разрешает выборку байта из ППЗУ, после чего выбранный байт поступает на порт P0 МК51 и вводится в ОМЭВМ.

Диаграммы, показывающие формирование соответствующих сигналов при работе ОМЭВМ с ВПП, приведены на рис.14.

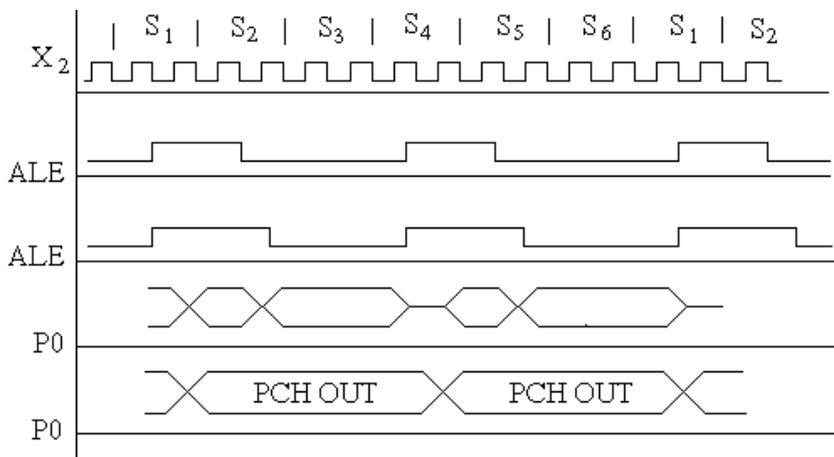


Рис.14. Диаграммы, показывающие формирование соответствующих сигналов при работе ОМЭВМ с ВПП

На рис.14. использованы следующие обозначения:

- PCL OUT - выдача младшего байта PC;
- PCH OUT - выдача старшего байта PC;
- INPUT - данные на шине P0.

Как видно из диаграмм, при работе с ВПП сигнал PМЕ формируется дважды в каждом машинном цикле независимо от количества байт в команде. Если второй выбираемый байт в текущей команде не используется, он игнорируется ОМЭВМ. В дальнейшем, при переходе к выполнению следующей команды этот байт будет введен вторично.

Структура микроконтроллеров семейства MCS-x96.

Семейство микроконтроллеров MCS-x96 предназначено специально для построения высокопроизводительных контроллеров, реализующих алгоритмы управления исполнительными двигателями постоянного и переменного тока.

Обобщенная структурная схема микроконтроллера семейства MCS-x96 изображена на рис. 16. В его состав входят процессор, память, набор периферийных устройств и контроллер памяти. К микроконтроллеру можно подключить внешнюю память.

Процессор содержит арифметико-логическое устройство (ALU) и регистровое оперативное запоминающее устройство (RRAM). Отличительная особенность ALU - отсутствие регистра-аккумулятора. При выполнении арифметических и логических операций в качестве источника первого операнда и приемника результата может использоваться любой регистр в RRAM, при этом операнд и результат могут иметь разные адреса. ALU обращается к RRAM непосредственно или через контроллер памяти.

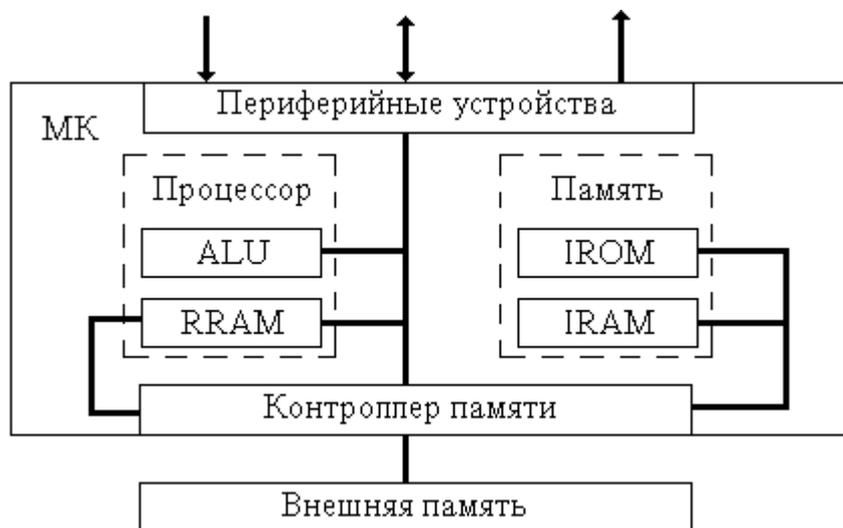


РИС.18. ОБОБЩЕННАЯ СТРУКТУРНАЯ СХЕМА МИКРОКОНТРОЛЛЕРА СЕМЕЙСТВА MCS-X96

Внутренняя (Internal) память микроконтроллера содержит постоянное (IROM) и оперативное (IRAM) запоминающие устройства. Первое из них используется для хранения команд программы, констант и специальных данных. Выпускаются микроконтроллеры, в которых IROM отсутствует (в этом случае его функции возлагают на запоминающее устройство, входящее в состав внешней памяти).

IRAM используется для хранения данных и команд программы. При этом открывается возможность модифицировать команды в процессе выполнения программы. В микроконтроллерах некоторых типов IRAM отсутствует.

Общее число адресов в адресном пространстве микроконтроллера равно 64К, а у микроконтроллеров подсемейства NT может быть увеличено до 1М.

Контроллер памяти управляет процессом обращения к внутренней и внешней памяти, при этом обеспечивается опережающая выборка кодов команд с образованием их очереди.

В таблице 3. указана емкость RRAM, IROM и IRAM микроконтроллеров разных типов, а также указаны интегрированные на кристалле микроконтроллера периферийные устройства. Буква X в обозначении типа заменяется цифрой 0, если микроконтроллер не имеет IROM, цифрой 3, если он имеет IROM масочного типа, и цифрой 7, если микроконтроллер содержит программируемое постоянное запоминающее устройство с возможностью стирания записи путем ультрафиолетового облучения (EPROM). При отсутствии окна в корпусе микросхемы возможно лишь однократное программирование IROM (OTPROM).

Таблица 3

МИКРО КОНТР ОЛЛЕР	8XC196KB	8XL196KD	8XC196NT	8XC196MC	8XC196MD
RRAM БАЙТ	232	1000	1000	488	488

ИРОМ КБАЙТ	8	32	32	16	16
ИРАМ БАЙТ	-	-	512	-	-
P	5	5	8	7	8
I/O	40	40	56	53	64
SLP	-	-	+	-	-
SP	+	+	+	-	-
SSIO	-	-	+	-	-
HSIO	+	+	-	-	-
EPA	-	-	+	+	+
ADC	8	8	4	13	14
PWM	1	3	-	2	2
WG	-	-	-	+	+
FG	-	-	-	-	+
PTS	-	+	+	+	+
OFD	-	-	+	-	-

Периферийные устройства микроконтроллеров семейства MCS-x96.

Расположенные на кристалле микроконтроллера периферийные устройства предназначены для приема и выдачи данных, ввода и вывода событий и аналоговых сигналов, обслуживания запросов прерывания и контроля правильности работы микроконтроллера. В таблице 3 приведены данные о наличии определенных периферийных устройств у микроконтроллеров разных типов и даны некоторые характеристики этих устройств.

Для приема и выдачи данных в параллельном коде используются параллельные порты. Микроконтроллеры разных типов имеют разное число таких портов, при этом разные порты могут иметь разное число разрядов. В графе P таблицы указано число параллельных портов у микроконтроллеров разных типов, а в графе I/O - суммарное число их разрядов.

Для обмена данными между микроконтроллером и центральным процессором в иерархической микропроцессорной системе предназначен процессорный порт (SLP - Slave Port), который подключают непосредственно к системной магистрали центрального процессора. В качестве SLP используется один из параллельных портов, который переводится в соответствующий режим путем программирования.

Для приема и выдачи данных в последовательном коде используется последовательный порт (SP). Он позволяет увеличивать число параллельных портов микроконтроллера путем подключения внешних сдвигающих регистров, обмениваться данными с другими устройствами по последовательному каналу связи (например, по интерфейсу RS - 232) и создавать простейшие локальные сети микроконтроллеров.

Для обмена данными в последовательном коде между двумя микроконтроллерами предназначен синхронный последовательный порт (SSIO). При этом могут использоваться две, три или четыре соединительные линии.

Все микроконтроллеры семейства MCS-x96 оснащены специальным периферийным устройством, предназначенным для приема и регистрации входных событий и формирования и выдачи выходных. Событием является изменение значения сигнала. Различают единичные события (замена нулевого значения единичным) и нулевые (замена единичного значения нулевым). Прием и регистрация входного события заключается в запоминании времени появления события определенного типа на определенном входе микроконтроллера. Это позволяет определять временные параметры входных импульсных последовательностей (период следования и длительность импульсов и т.д.).

При формировании и выдаче выходного события в определенное, заранее заданное время появляется событие определенного типа на определенном выходе микроконтроллера (внешнее выходное событие) или в определенной точке внутри микроконтроллера (внутреннее выходное событие). Это позволяет формировать импульсные последовательности заданной формы (например, широтно-модулированный сигнал) и реализовывать временные задержки.

Для работы с событиями в микроконтроллерах подсемейств KB и KC используется блок быстрого ввода-вывода (HSIO), а в микроконтроллерах подсемейств KR, NT, MC - блок процессоров событий (EPA). В HSIO имеется определенное число входных и выходных каналов, в EPA - универсальные модули, каждый из которых может быть запрограммирован для работы с входными или выходными событиями.

Для ввода аналоговых сигналов используется многоканальный аналогово-цифровой преобразователь (ADC). Значение аналогового сигнала представляется восьми- или десятиразрядным двоичным кодом. Число каналов в преобразователе указано в графе ADC.

Микроконтроллеры подсемейств KB, KC, MC имеют широтно-импульсный модулятор (PWM) с программируемым значением скважности импульсной последовательности. Использование этого модулятора совместно с внешним интегрирующим устройством позволяет осуществлять цифроаналоговое преобразование.

Микроконтроллеры подсемейства MC содержат трехфазный генератор (WG - Waveform Generator), который может быть использован для управления трехфазными электродвигателями переменного тока, вентильными электродвигателями постоянного тока, шаговыми двигателями, а также для преобразования постоянного тока в переменный.

В микроконтроллерах 8XC196MD имеется генератор меандра (FG-Frequency Generator) с программируемой длительностью импульсов и периодом их следования.

Все микроконтроллеры семейства MCS-x96 имеют систему управления прерываниями. С ее помощью осуществляется переход от выполнения текущей программы к выполнению прерывающей, составленной программистом и записанной в память микроконтроллера. Для обслуживания прерываний

микроконтроллеры подсемейств КС, КR, NT, МС имеют, кроме того, периферийный сервер транзакций (PTS - блок обслуживания групповых операций).

Обслуживание запроса прерывания с использованием PTS заключается в выполнении вместо очередной команды текущей программы определенной микропрограммы, заложенной в специальную память микроконтроллера при его изготовлении. Программист выбирает для обслуживания запроса прерывания подходящую микропрограмму из набора имеющихся и настраивает ее для выполнения в каждом конкретном случае путем записи группы кодов в RRAM.

В число операций, которые могут быть выполнены под управлением PTS при обслуживании запроса прерывания, входят одиночная и групповая пересылки и чтение результатов аналого-цифрового преобразования. Кроме того, микроконтроллеры подсемейства КС могут выполнять операции, связанные с регистрацией входных и формированием выходных событий, микроконтроллеры подсемейств КR, NT-операции, связанные с формированием широтно-модулированных сигналов, а подсемейства МС - операции, реализующие функции последовательного порта.

Для контроля правильности работы все микроконтроллеры оснащены сторожевым таймером, сбрасывающим их в исходное состояние при появлении сбоя в ходе программы. Микроконтроллеры подсемейств КR, NT, кроме того, содержат детектор падения частоты (OFD), который переводит их в состояние сброса при катастрофическом снижении тактовой частоты.

Лекция 10.

Преимущества регистр-регистровой архитектуры

При построении большинства микропроцессоров и микроконтроллеров (например, серии 8-разрядных микроконтроллеров MCS-51) используется традиционная, так называемая *аккумуляторная архитектура*, когда один из регистров специального назначения *аккумулятор* по умолчанию является источником одного из двух операндов и одновременно приемником результата операции, выполняемой в *арифметико-логическом устройстве* (АЛУ). Такой подход позволяет существенно уменьшить формат команды за счет адресации в поле операндов только одного из операндов.

Для выполнения какого-либо действия в процессорах «классической» аккумуляторной архитектуры в общем случае кроме собственно команды, пример, сложения, требуются по крайней мере еще две операции пересылки данных: загрузка первого операнда в аккумулятор и через него в один из портов арифметико-логического устройства (АЛУ), а также пересылка результата операции из аккумулятора по месту назначения уже после завершения операции (рис.17) при этом второй операнд извлекается из оперативной памяти непосредственно в процессе выполнения операции. Также в процессе

выполнения операции результат помещается в аккумулятор.

По мере развития микропроцессорной техники аккумуляторная архитектура стала тормозом в повышении производительности центрального процессора и уступила место так называемой *регистр-регистровой архитектуре*. Основным отличием регистр-регистровой архитектуры от аккумуляторной является то, что любые ячейки интегрированной на кристалл микроконтроллера сверхбыстродействующей оперативной памяти, *регистрового ОЗУ*, могут рассматриваться либо как операнды-источники, либо как аккумулятор для приема результата операции. При таком подходе любая из команд процессора может быть трехоперандной и необходимость в дополнительных операциях пересылки данных из аккумулятора и обратно отпадает. Естественно, что регистр-регистровая архитектура требует более сложного формата команды. Наряду с полем кода операции в формате команды будут присутствовать поля адресов операндов-источников и операнда-приемника.

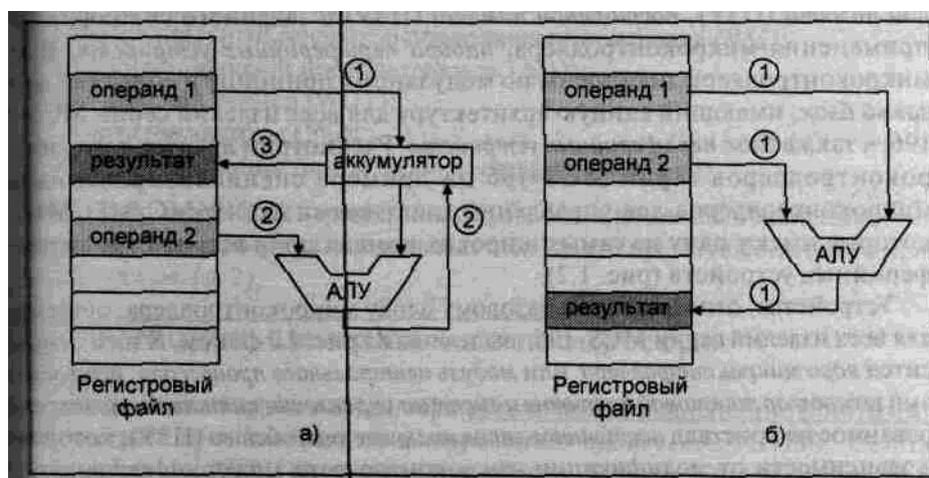


РИС. 17. СРАВНЕНИЕ АККУМУЛЯТОРНОЙ (А) И РЕГИСТР-РЕГИСТРОВОЙ АРХИТЕКТУРЫ (Б)

Таким образом, любая *ячейка регистрового файла*, внутренней сверхоперативной памяти, может выполнять функцию как источника данных, так и приемника данных.

Главные преимущества регистр-регистровой архитектуры:

- Отсутствует необходимость в предварительной загрузке одного из операндов в аккумулятор перед выполнением операции.
- Содержимое источников данных в результате выполнения операции может оставаться неизменным, либо модернизироваться по желанию программиста. В последнем случае источник данных одновременно будет и приемником результата операции.
- Резко уменьшается число операций по пересылке данных и возрастает скорость вычислений.
- Появляется возможность расширения системы команд трехоперандными высокопроизводительными командами.

Совокупность перечисленных выше факторов вместе с главным преимуществом - возможностью выполнения операций непосредственно над 16-разрядными операндами, приводит к повышению производительности 16-разрядных микроконтроллеров *в несколько раз* по сравнению с 8-разрядными контроллерами, работающими на тех же тактовых частотах.

Блок-схема микроконтроллера MCS-196.

Микроконтроллеры семейства MCS-196 являются однокристальными микроЭВМ с интеграцией на кристалл центрального процессора, оперативной памяти (ОЗУ), постоянной памяти (ПЗУ) и заданного спецификой применения микроконтроллера, набора периферийных устройств. Все микроконтроллеры построены по модульному принципу и содержат базовый блок, имеющий единую архитектуру для всех изделий серии MCS-196, а также блок периферийных устройств. Рассмотрим архитектуру микроконтроллеров серии MCS-196 на примере специализированных микроконтроллеров для управления двигателями 8xC196MC/MD/MH, которые имеют одну из самых широких номенклатур встроенных периферийных устройств (рис. 18).

Устройства, относящиеся к базовому блоку микроконтроллера, общему для всех изделий серии MCS-196, выделены на рис. 18 фоном. К ним относится ядро микроконтроллера или модуль центрального процессора, встроенный генератор тактовой частоты и система управления питанием, интегрированное на кристалл постоянное запоминающее устройство (ПЗУ), которое в зависимости от модификации микроконтроллера может отсутствовать. контроллер прерываний (Interrupt Controller), а также сервер периферийных транзакций (PTS) — специализированный контроллер прерываний, обеспечивающий быстрый обмен данными между периферийными устройствами и памятью практически без участия центрального процессора.

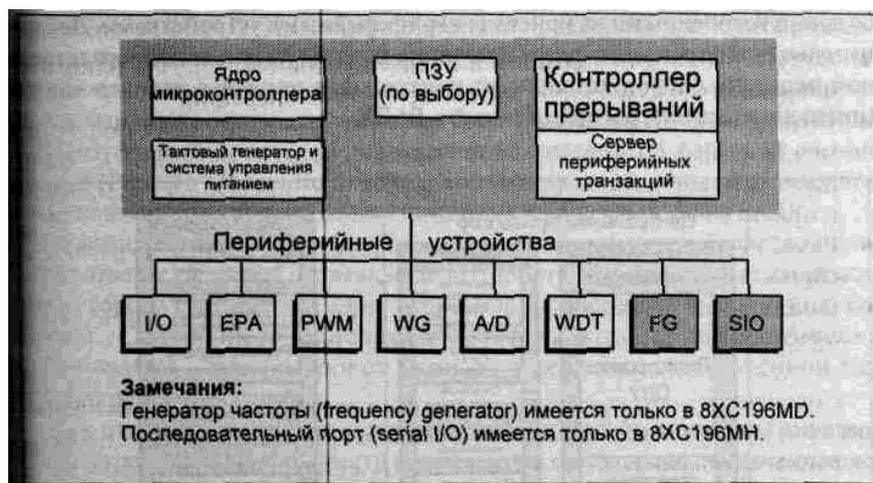


Рис. 18. Типовая архитектура микроконтроллеров семейства MCS-196 на примере микроконтроллеров для управления двигателями 8xC196MC/MD/MH.

В набор интегрированных на кристалл периферийных устройств для микроконтроллеров 8xC196MC/MD/MH входят:

порты ввода/вывода (I/O);

процессор событий (ЕРА);
 генератор широтно-импульсных сигналов (РWМ);
 многоканальный генератор периодических сигналов (WГ);
 аналого-цифровой преобразователь (А/D);
 сторожевой таймер (WDT);
 генератор частоты (FG1);
 последовательные порты ввода/вывода данных (SIO).

Набор встроенных периферийных устройств меняется в зависимости от типа микроконтроллера и определяет область его преимущественного применения.

Модуль центрального процессора. *Модуль центрального процессора обеспечивает выполнение программы пользователя и состоит из центрального процессора (ЦПУ) Central Processing Unit (CPU) и контроллера памяти Memory Controller (рис. 19). В состав центрального процессора входят регистровый файл Register File и регистровое арифметико-логическое устройство (Register Arithmetic-Logic Unit) или сокращенно РАЛУ (RALU).*

Внутренняя 16-разрядная шина связывает центральный процессор с контроллером памяти и контроллером прерываний (interrupt controller), а также со всеми встроенными на кристалл периферийными устройствами. Дополнительная внутренняя 8-разрядная шина используется для непосредственной передачи байтов команды из контроллера памяти в регистр команд (instruction register), расположенный в РАЛУ.

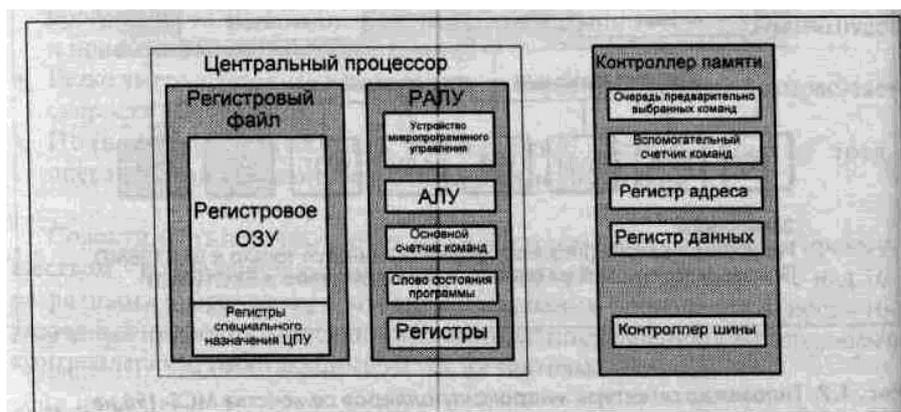


РИС. 19. СТРУКТУРА МОДУЛЯ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

В состав центрального процессора входит *устройство микропрограммного управления* (microcode engine), предназначенное для приема от контроллера памяти кода очередной команды, *дешифрации кода* и генерации последовательности управляющих команд, подаваемых на регистровое арифметико-логическое устройство, и необходимых для выполнения операций над байтами, словами и длинными словами, расположенными в 256-байтовом *нижнем регистровом файле*. Использование механизма адресации операндов через *вертикальное окно* в нижнем регистровом файле, позволяет выполнять в РАЛУ также и операции с данными, расположенными в верхнем регистровом файле, а в новейшей серии микроконтроллеров MCS-296 — даже во внешней памяти.

Центральный процессор связывается с внутренней кодовой и внешней памятью (программ и данных) через *контроллер памяти*. Исключение составляет регистровый файл, доступ к которому обеспечивается непосредственно. В состав контроллера памяти входят несколько устройств: *вспомогательный счетчик команд* (Slave PC), 4-байтовая *очередь* из предварительно выбранных из памяти и готовых для выполнения кодов команд (Queue), *контроллер шины* (Bus Controller), *регистры адреса* (Adress Register) и *данных* (Data Register).

Запрос на доступ к памяти может поступить как от центрального процессора при выполнении текущей команды, например, для получения операнда из внешней памяти по косвенному адресу, расположенному в одном из регистров-слов внутреннего ОЗУ, так и от диспетчера очереди предварительно выбранных кодов команд. Механизм предварительной выборки команд существенно повышает производительность центрального процессора так как к моменту завершения выполнения текущей команды следующая команда уже считана из памяти и готова для немедленного выполнения. Центральный процессор получает коды команд из 4-байтовой *очереди команд* (Prefetch Queue по специальной 8-разрядной шине. Код текущей исполняемой инструкции всегда находится в *регистре команд* РАЛУ. За выполнение программы пользователя в правильной последовательности отвечает *основной счетчик команд* (Master Program Counter), содержимое которого указывает на адрес очередной, подлежащей выполнению команды. Содержимое основного счетчика команд автоматически увеличивается в зависимости от длины текущей команды встроенной схемы инкрементирования.

Как отмечено выше для ускорения выборки кодов команд из внешней или внутренней кодовой памяти используется *вспомогательный счетчик команд* (Slave PC), содержимое которого задает адрес команды, подлежащей выборке из памяти. Выбранная команда поступает в очередь на выполнение. Как только очередная команда считывается из очереди и передается на дешифрирование в устройство микропрограммного управления центрального процессора, диспетчер очереди формирует запрос в *контроллер шины* Bus Controller) на считывание очередной команды. Контроллер шины входит в состав контроллера памяти и обеспечивает формирование необходимых управляющих сигналов как для считывания данных из внешней, внутренней памяти или внешнего периферийного устройства, так и для записи данных во внешнее ОЗУ или в периферийное устройство. Таким образом, контроллер шины обеспечивает считывание кодов машинных команд и запись/считывание данных. Запрос на считывание очередной инструкции поступает от диспетчера очереди, а запрос на запись/считывание данных (операндов) - от центрального процессора. Запрос от диспетчера очереди имеет более высокий приоритет.

Если линейная последовательность выполнения команд программы нарушается, например, вследствие выполнения команды условного или безусловного перехода или вызова подпрограммы, значение вспомогательного счетчика команд перезагружается текущим значением основного счетчика команд и очередь ранее выбранных команд аннулируется (очищается). Перезагрузка вспомогательного счетчика команд и извлечение первого байта

кода инструкции из памяти занимает 4 машинных такта.

Замечание. Если при отладке микропроцессорной системы Вы пользуетесь логическим анализатором, то имейте в виду, что команды из памяти выбираются с опережением. В данный момент выполняется команда, которая уже была считана ранее

Регистровое АЛУ имеет достаточно сложную внутреннюю структуру. В его состав помимо собственно 16-разрядного арифметико-логического устройства, регистра команд и устройства микропрограммного управления входят несколько вспомогательных регистров:

- Регистры старшего и младшего слова, предназначенные для использования в командах с 32-разрядными операндами в качестве регистров временного хранения данных. Они имеют собственные устройства сдвига, что ускоряет выполнение команд сдвига, нормализации, умножения и деления.

- Регистр второго операнда АЛУ, используемый в двухоперандных командах. в частности, для хранения множителя или делителя в операциях умножения, деления, соответственно.

- 6-разрядный счетчик числа циклов для организации сдвига операнда на заданное число разрядов со схемой автоматического декрементирования.

- Регистр селектирования заданного бита, который позволяет автоматически генерировать необходимую маску при выполнении команд тестирования бита по номеру бита, указанному в команде.

- Регистры констант (0, 1, 2), необходимых для ускорения типовых операций в АЛУ.

После завершения любой операции в АЛУ результат анализируется и соответствующие *признаки результата* (нуля, переноса, переполнения и т.д.) выставляются в *слове состояния программы* Program Status Word (PSW). Признаки результата операции используются в командах условной передачи управления для организации ветвлений в программе.

Регистровый файл

Различные типы микроконтроллеров отличаются количеством ячеек памяти в регистровом файле, так называемом *регистровом ОЗУ* (РОЗУ). Естественно, что чем больше объем сверхоперативной регистровой памяти, тем более сложные вычисления и с большим быстродействием могут быть организованы в микропроцессорной системе на базе данного микроконтроллера.

Первые 256 ячеек регистрового файла имеют короткие байтовые адреса и относятся к *нижнему регистровому файлу*, а все остальные — 16-разрядные адреса и относятся к *верхнему регистровому файлу*. Наибольшим быстродействием обладают команды с прямой регистровой адресацией операндов в нижнем регистровом файле. При этом используются байтовые адресные поля и команды имеют предельно короткий формат и наименьшее время выполнения. Для доступа к операндам за пределами нижнего

регистрового файла (в том числе и к внешней памяти) используются специальные способы адресации, в частности косвенная, базово-индексная, а также адресация через специальные вертикальные окна.

Младшие 24 байта нижнего регистрового файла относятся к *регистрам специального назначения центрального процессора* (CPU special-function registers - SFRs). Здесь расположен также *указатель стека* (Stack Pointer SP). Вся остальная область регистрового ОЗУ, в том числе область верхнего регистрового файла, является оперативной памятью общего назначения и может использоваться программистом по своему усмотрению. Доступ производится к байтам, словам или двойным словам. Набор регистров специального назначения отличается в микроконтроллерах разных типов. Эти регистры позволяют управлять системой прерываний, селективировать вертикальные окна и выполнять целый ряд дополнительных функций.

Особенность организация памяти микроконтроллеров MCS-196

Типичная схема организации памяти 16-разрядных микроконтроллеров Intel на примере 8XC196KC и 8XC196KD приведена на рис. 20.

Нижняя область встроенного регистрового ОЗУ (0000H-0017H) жестко закреплена за *регистрами специального назначения центрального процессора (CPU SFRs)*, через которые осуществляется управление системой прерываний, механизм селективирования горизонтальных и вертикальных окон, а также встроенными периферийными устройствами (для некоторых типов микроконтроллеров).

Начиная с адреса 0018H располагается *область регистрового ОЗУ (Register RAM)*, объем которой различен для разных контроллеров и варьируется как правило, в диапазоне от 232 байт до 1 Кбайта.

Следующий сектор памяти объемом несколько меньше 8 Кбайт используется для адресации *внешней памяти и/или внешних периферийных устройств*. В конце этой области памяти располагается еще одна область регистров специального назначения - *регистров специального назначения периферийных устройств SFR* (в данном случае регистров портов ввода/вывод 10PORT3 и 10PORT4). Размер этой области определяется объемом интегрированной на кристалл периферии и, например, у контроллеров 8XC196MC/MD/MH достигает 256 байт (1FE0H-1FFFH).

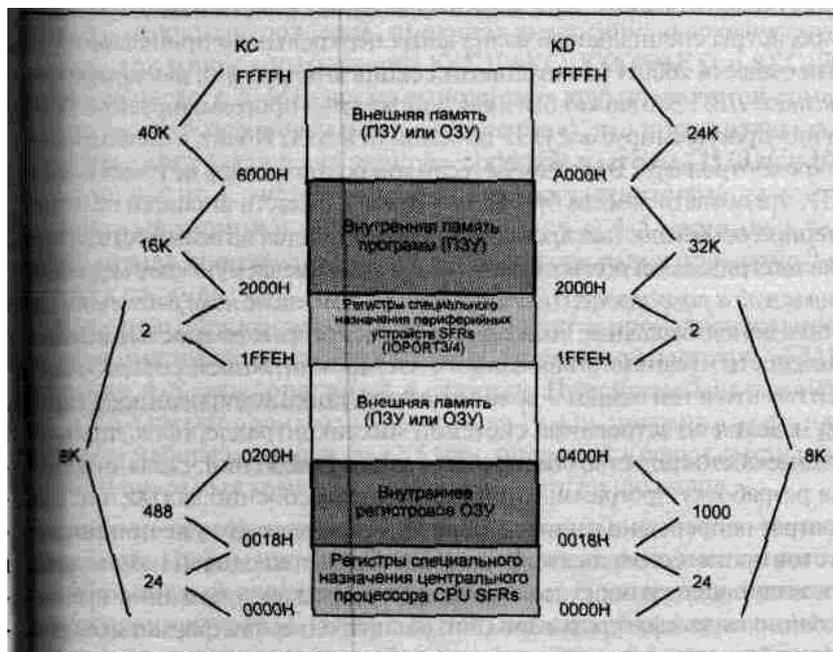


Рис 20. Карта распределения памяти для микроконтроллеров 8x86C196KC/KD

Дело в том, что в микроконтроллерах с большим количеством интегрированных на кристалл периферийных устройств начальной области памяти 24 байта оказывается недостаточно для размещения всех регистров специального назначения периферийных устройства для их размещения отведена специальная область памяти. При этом в начальной области памяти остаются только регистры специального назначения центрального процессора.

Начиная с адреса 2000H располагается секция *внутреннего, интегрированного на кристалл ПЗУ*. Это может быть как однократно-программируемое ПЗУ так и масочно-программируемое ПЗУ объемом от 8 до 32 Кбайт, в зависимости от типа микроконтроллера. В том случае, если микроконтроллер не имеет встроенного ПЗУ, эта область памяти будет относиться к области внешней памяти.

Характерной особенностью процессоров Intel является возможность задать в начале рассматриваемой области памяти не только желаемую *конфигурацию микроконтроллера и микропроцессорной системы*, в том числе *цикл работы внешней шины*, но и дополнительные *коды секретности*, которые полностью исключают возможность несанкционированного чтения или записи содержимого внутренней памяти и тем самым — возможность несанкционированного тиражирования изделия со встроенной системой микроконтроллерного управления. Последнее обстоятельство обеспечивает защиту вложений, сделанных заказчиком в разработку программного продукта, что особенно важно, так как доля этих затрат непрерывно растет и по некоторым оценкам уже приближается к 80% стоимости всего проекта. В начальной области кодовой памяти располагаются также вектора переходов на подпрограммы обслуживания прерываний.

Первым адресом, по которому процессор обращается для выполнения программы пользователя, является адрес 2080H. Сюда передается управления и

после сброса системы по сигналу RESET. Расположенная вслед за внутренним ПЗУ память, вплоть до адреса FFFFH может рассматриваться как *внешняя память программ и/или данных и/или область адресации внешних устройств ввода/вывода пользователя*. Некоторые самые современные микроконтроллеры, например, 8XC196NP, 8XC196NU, 80C296SA, предназначенные для высокопроизводительной обработки больших массивов информации в реальном времени имеют *возможности расширенной адресации памяти*. В этом случае объем прямоадресуемой памяти возрастает до 1 Мбайта, 6 Мбайт и даже до 16 Мбайт. При этом внутренняя шина адреса микроконтроллера вместо 16-разрядной становится 24-разрядной, а во вне выводятся дополнительно 4 или 8 старших разрядов адресной шины.

Возможности подключения внешней памяти и внешних устройств ввода/вывода

На стадии разработки и отладки микропроцессорной системы приходится использовать *внешнюю память программ* и только затем, когда алгоритмы и программное обеспечение полностью оттестированы и начинается серийный выпуск изделия, программное обеспечение «прошивается» в внутреннюю память микроконтроллера. Кроме того, в ряде применений требуется *внешняя память данных*, часто энергонезависимая, для хранения промежуточных массивов информации, результатов тестирования работоспособности оборудования, текущих настроек системы управления, циклограмм управления оборудованием в функции времени и т.д.

В системах встроенного управления, где часто приходится изменять алгоритмы управления и программное обеспечение, единственной альтернативой оказывается применение в качестве внешней памяти программ так называемой *флэш-памяти*. Эта память, впервые анонсированная фирмой Intel в 1988 г., бурно развивается и на сегодняшний день сочетает в себе достоинства ОЗУ и ПЗУ, являясь энергонезависимой памятью с высокой плотностью хранения информации, высоким быстродействием, малым потреблением энергии, исключительной надежностью и помехоустойчивостью. Она допускает более 100 тысяч циклов перезаписи.

Вместе с тем, несмотря на большое количество интегрированных на кристалл периферийных устройств, их часто оказывается недостаточно для решения конкретной задачи. В этом случае приходится дополнительно использовать *внешние периферийные устройства*: программируемые порты ввода/вывода данных, контроллеры прерываний, контроллеры последовательного ввода/вывода данных и т.д.

Встроенный контроллер внешней шины обеспечивает корректное подключение к микроконтроллеру внешней памяти и внешних периферийных устройств, а также формирование необходимых сигналов управления ими. Большинство микроконтроллеров MCS-196 имеют *мультиплицированную шину адреса/данных*, а самые совершенные из них могут работать и с *мультиплицированной шиной адреса/данных*, что повышает

производительность системы и упрощает интерфейс с внешней памятью и периферией.

В зависимости от разрядности используемой внешней памяти и периферии шина может работать в нескольких режимах:

Стандартном — как 16-разрядная мультиплицированная шина (вся внешняя память и периферия 16-разрядные). *8-разрядном* — когда вся внешняя память и периферия 8-разрядные. *Динамически переключаемом* между 8- и 16-разрядным режимами в зависимости от типа внешнего устройства, к которому в данный момент происходит обращение.

Кроме того, наличие специального входа готовности Ready допускает расширение сигналов чтения RD# и записи WR# для подключения медленнодействующей памяти и устройств ввода/вывода (например, жидкокристаллического дисплея). Более того, изменяя программно *байты конфигурации микроконтроллера*, можно задать требуемое число тактов ожидания для оптимального сопряжения микроконтроллера с внешними устройствами.

Лекция 11.

Применение микропроцессоров в приводах мехатронных систем. Примеры МП приводов современных систем.

Пример МК-системы на основе ОМЭВМ семейства МК51.

В данном разделе рассматривается пример построения МК-системы на основе ОМЭВМ МК51, который может быть использован для приобретения навыков программирования и отладки программ для МК51, а также рассматриваться в качестве макета реальной системы управления для отладки ее прикладного программного обеспечения совместно с объектом управления в реальном масштабе времени.

Проектируемый отладочный модуль должен содержать следующие средства:

- постоянную память с программой Монитор, обеспечивающей управление всей системой и прежде всего взаимодействие с оператором;

- оперативную память для занесения программ, данных и создания буферов Монитора;

- простейшую клавиатуру для загрузки кодов и взаимодействия с оператором: 16 цифровых клавиш (цифры от 0 до F), 8 управляющих клавиш, в том числе и клавиша "сброс" - системного сброса;

- простой дисплей, обычно однострочный (на основе семисегментных индикаторов), который используется для визуального контроля вводимой информации и отображения данных;

- расширитель ввода-вывода на основе ППА КР580ВВ55.

Кроме того, отладочный модуль может иметь собственный источник

электропитания и (в некоторых случаях) дополнительное место на печатной плате ("слот") для монтажа пользовательских средств связи с объектом управления. Функциональная схема отладочного модуля, удовлетворяющего поставленным требованиям, приведена на рис.15. Как видно из приведенной схемы, память МКС имеет две физически-разделенные области для хранения данных и программ: ВПД и ВПП.

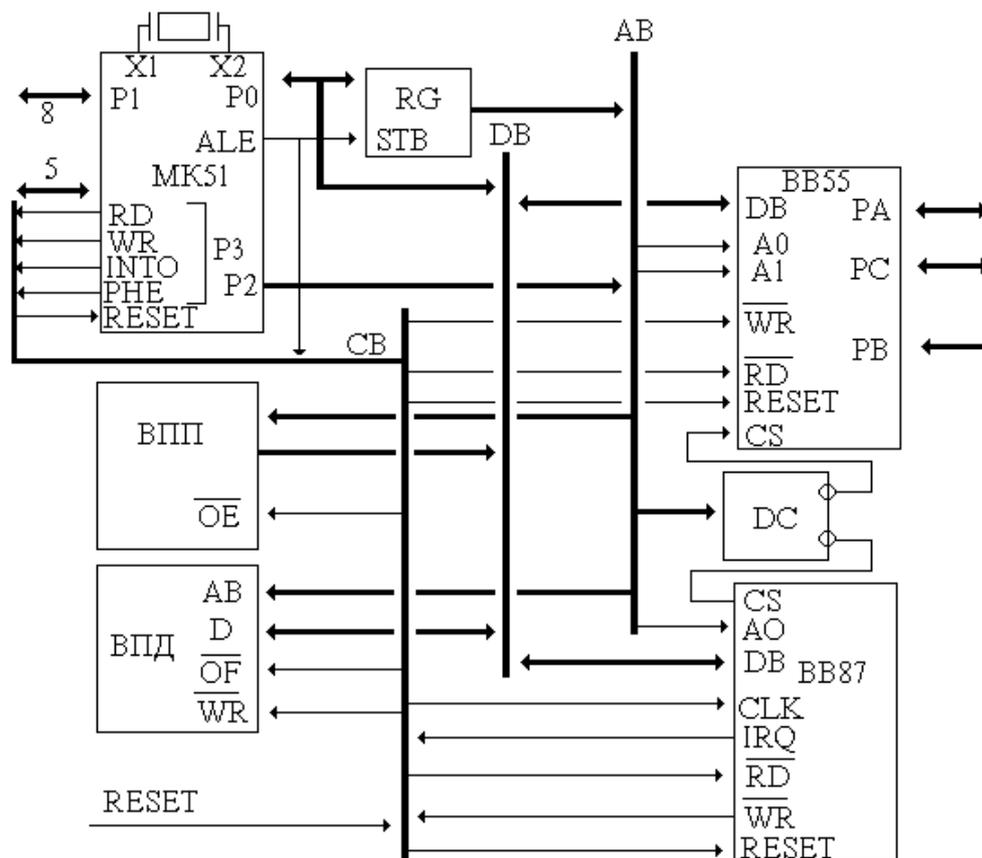


Рис.15. Функциональная схема отладочного модуля

Обращение к ним строится сигналами RD, WR и PSEN, соответственно. Программа Монитор расположена в ВПП. Адресное пространство ВПД включает в себя следующие области: буферную область Монитора; память данных; область, отведенную для подключения интерфейсных БИС (ППА, ККД).

Контроллер клавиатуры - дисплея (ККД) закрывает адресные пространства ВПД размером в 4 ячейки. Выбор кристалла ККД осуществляется дешифратором ДС, при установке соответствующих адресов (указанных 4-х ячеек).

Обмен данными МК с ККД осуществляется через порт P0. Для синхронизации работы ККД используется сигнал с выхода ALE. Выход сигнала прерывания из ККД (IRQ) соединен со входом запроса прерываний МК. Это позволяет достаточно просто обнаруживать факт нажатия клавиш, а также выполнять переход из программы пользователя в Монитор по прерыванию от клавиатуры.

Для управления внешним объектом от МК используется порт P1 и

оставшиеся не задействованными 5 линий порта P3. Для сопряжения МК с объектом, имеющим большое число входов/выходов, можно расширить резидентную систему ввода/вывода, подключив к МК необходимое количество внешних портов. Такое расширение может быть выполнено двумя способами: с использованием стандартного расширителя ввода/вывода (KP580BP43) или интерфейсных БИС (KP580BB55, KP580BB51). На приведенной схеме показан вариант расширения ввода/вывода с использованием параллельного периферийного адаптера (ППА) KP580BB55. Порты адаптера адресуются как ячейки ВПД, при этом младшие разряды (A0, A1) шины адреса используются для выбора порта, а старшие (A10 ÷ A3) с помощью дешифратора DC формируют сигнал выбора микросхемы ППА.

Ввод/вывод данных стробируется сигналами WR/RD - записи/чтения ВПД. Системный сброс МКС осуществляется нажатием на клавишу "сброс".

Унифицированная структура силовой части статических преобразователей частоты для широкого класса приводов переменного тока.

Современный этап технической революции характеризуется бурным развитием не только управляющей, но и силовой электроники, успехи которой позволяют по новому взглянуть на ряд проблем, которые раньше не решались либо ввиду непомерной стоимости проекта либо сложности его реализации на имеющейся элементной базе.

Известно, что в большинстве промышленно развитых стран около 70-80% всей вырабатываемой электроэнергии используется для преобразования в механическую работу с помощью различных электродвигателей. При этом до последнего времени наиболее массовым типом электропривода был нерегулируемый привод с асинхронными короткозамкнутыми двигателями. Это объясняется простотой двигателя и возможностью его непосредственного подключения к трехфазной сети переменного тока с помощью так называемых пускателей - простейшей коммутационной и защитной аппаратуры.

Применение нерегулируемых электроприводов сопровождается значительными непроизводительными затратами электроэнергии и в условиях резкого подорожания энергоносителей и, соответственно, электроэнергии, экономически невыгодно. Поэтому, доля регулируемых электроприводов, в том числе частотно-регулируемых электроприводов переменного тока постоянно растет. При этом в ряде стран приняты специальные государственные программы ускоренного развития энерго- и ресурсосберегающих технологий на базе экономичного регулируемого электропривода с широкими интерфейсными возможностями, позволяющими одновременно с экономией электроэнергии решать и задачи комплексной автоматизации производства в целях оптимизации параметров технологических процессов и улучшения качества выпускаемой продукции.

Увеличение стоимости энергоносителей привело к еще одной проблеме - удорожанию цветных металлов и к увеличению стоимости электрических машин традиционных конструкций. Эта проблема способствовала быстрому прогрессу электрических машин новых типов с малыми затратами меди и

алюминия («малообмоточных»): реактивных и индукторных, которые к тому же более технологичны, так как имеют не распределенные, а более простые в изготовлении сосредоточенные обмотки.

В отличие от асинхронных двигателей эти двигатели не могут работать непосредственно от сети переменного тока и требуют питания от *инвертора напряжения или тока* с регулируемой выходной частотой. Особенно эффективна работа таких машин в режиме *автокоммутации*, когда переключение фаз выполняется по сигналам установленного на вал двигателя (встроенного) датчика положения. Такой двигатель называется *вентильным* или *бесконтактным двигателем постоянного тока* и допускает управление по двум каналам: каналу напряжения на якоре и каналу угла коммутации. По прогнозам специалистов вентильные двигатели в начале следующего века вытеснят с рынка как традиционные коллекторные двигатели постоянного тока, так и асинхронные двигатели.

Большинство современных систем привода переменного тока со статическими преобразователями частоты строится по хорошо известной классической схеме: «Неуправляемый выпрямитель - Звено постоянного тока - Автономный инвертор напряжения с ШИМ-модуляцией - Двигатель» (рис.16). Для трехфазных двигателей (асинхронных, синхронных, вентильных или вентильных индукторных) инвертор напряжения выполняется по стандартной 6-и ключевой схеме. Для двигателей с другим числом фаз, например, для четырехфазных индукторных двигателей инвертор строится на базе полумостовых или мостовых схем.

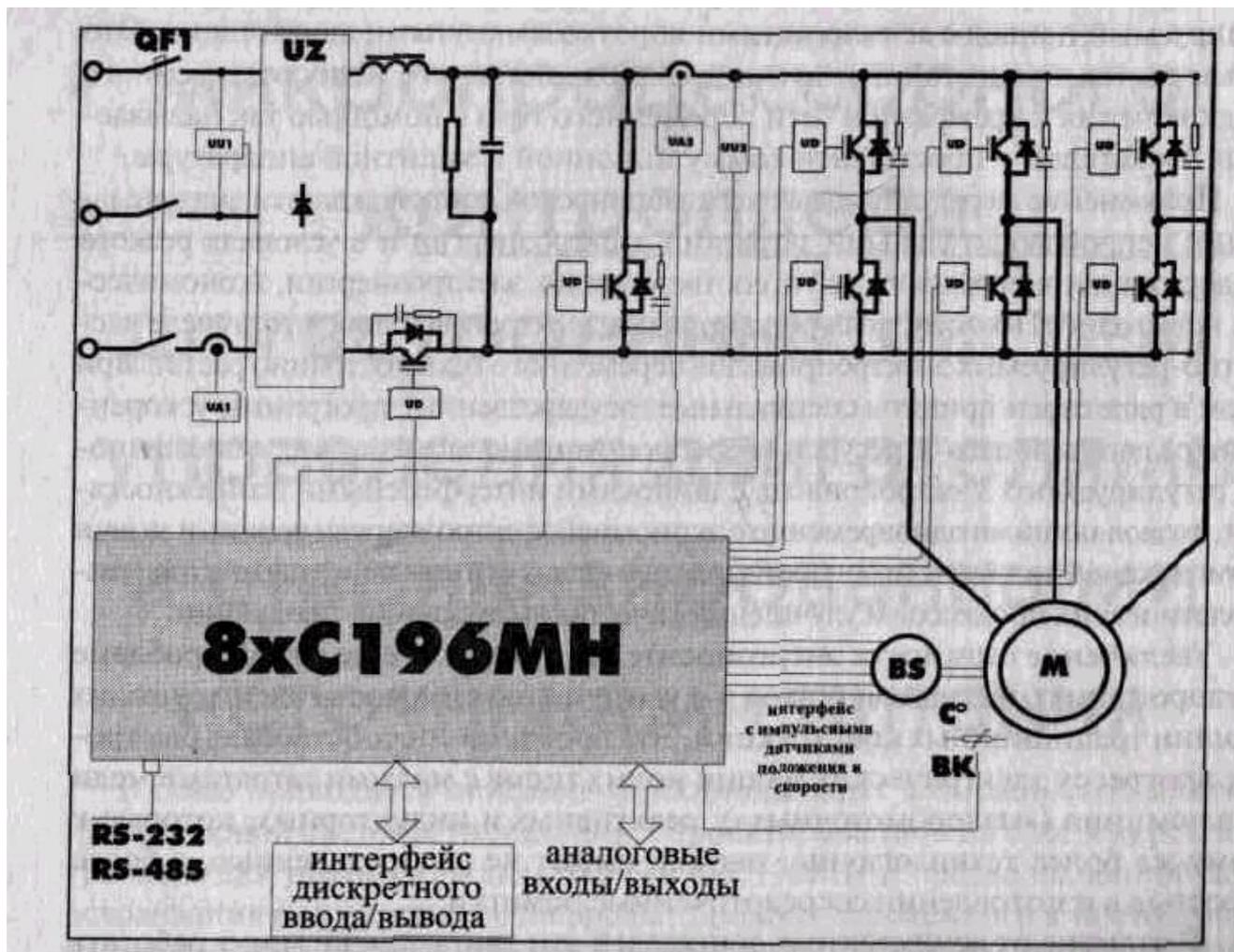


РИС. 16. ТИПОВАЯ СТРУКТУРА ПРИВОДА ПЕРЕМЕННОГО ТОКА

Силовой преобразователь состоит из неуправляемого выпрямителя UZ, подключаемого к однофазной или трехфазной сети переменного тока и автономного инвертора напряжения, работающего, как правило, в режиме синусоидальной широтно-импульсной модуляции. К выходу инвертора подключен двигатель M (асинхронный или вентильный постоянного тока). В последнем случае на вал двигателя дополнительно устанавливается датчик положения BS.

Инвертор может быть построен с использованием *биполярных транзисторов с изолированным затвором IGBT*. В широком диапазоне токов и напряжений рядом фирм выпускаются как отдельные силовые транзисторы, так и стойки, моду; и и даже интеллектуальные модули, имеющие встроенные драйверы и элементы защиты, а также интерфейсы, допускающие непосредственное подключение микропроцессорных систем управления. Имеются также и другие необходимые для построения привода компоненты: электролитические конденсаторы на высокие напряжения и токи, диодные модули для построения выпрямителей и т.д.

Мощная, встроенная в преобразователь микропроцессорная система управления должна выполнять функции не только прямого цифрового управления инвертором напряжения, но и всеми дополнительными элементами преобразователя, а именно:

- Цепью возбуждения для синхронных и вентильных двигателей с независимым контуром возбуждения (который физически может располагаться как на роторе, так и на статоре).

- Цепью защиты преобразователя от перенапряжений при рекуперации энергии (путем управляемого «слива» энергии в дополнительный балластный резистор) для приводов с активной нагрузкой на валу или приводов с высокими требованиями по динамике переходных процессов.

- Цепью заряда батареи конденсаторов в контуре постоянного тока при включении преобразователя в работу.

- Цепью питания электромагнитного тормоза (если он необходим).

Кроме того, встроенная система микропроцессорного управления должна напрямую обрабатывать сигналы ряда датчиков обратных связей, среди которых типовыми являются датчики напряжения и тока в цепи постоянного тока, тока возбуждения двигателя, температуры двигателя и (или) преобразователя, а также датчики положения ротора различных типов (на базе чувствительных элементов Холла, оптических импульсных датчиков положения и т.д.), необходимые для выполнения функций автокоммутации в приводах с вентильными двигателями. Датчики положения или скорости, например, импульсные, могут потребоваться также в приводах с асинхронными двигателями целях более точной компенсации скольжения в функции нагрузки для расширения диапазона регулирования. Таким образом, встроенная микропроцессорная система управления должна иметь развитый интерфейс как с источниками аналоговых сигналов, так и с источниками импульсных сигналов.

Отметим, что современная встроенная система управления преобразователем частоты не будет конкурентоспособной без интерфейса с системами более высокого уровня управления (последовательного интерфейса RS-232 или 485), а также интерфейса с пультом оперативного управления, предназначенным для организации интерактивного взаимодействия с оператором для настройки параметров преобразователя, выбора нужного режима работы, наблюдения за координатами привода и технологическими переменными и т.д.

Сотни различных фирм, среди них такие известные как ABB, Siemens, Alien Bradley, Dan bss и др., выпускают преобразователи частоты и комплектные электропривода на их основе в диапазоне мощностей от 100 ВА до 100 кВА, работающие с высоким коэффициентом полезного действия до 96% и коэффициентом мощности до 0.9.

Серьезный научный потенциал, накопленный в России в области теории автоматического управления и теории электропривода, теории синхронных, шаговых и вентильных двигателей, вместе с доступностью современных микроэлектронных компонент (как силовых, так и управляющих), позволяет быстро выполнять необходимые разработки и запускать в производство отечественные серии преобразователей, основные технические характеристики которых соответствуют мировому уровню.

Приведем результаты проектирования универсального микропроцессорного контроллера для систем встроенного управления широкой гаммой приводов переменного тока со статическими преобразователями частоты, а также стабилизированными источниками питания.

Универсальный контроллер для встроенных применений

На рис. 17. показана блок-схема системы управления для использования в частотно-регулируемых электроприводах с асинхронными, синхронными, вентильными и шаговыми двигателями, а также в системах вторичного стабилизированного питания. Диапазон применений системы управления широк: от приводов насосов-дозаторов жидких сред до двухпроцессорных систем векторного управления приводами стержней ядерных реакторов.

При проектировании система управления была принята концепция совмещения функций прямого цифрового управления элементами преобразователя частоты и технологическими переменными, значение которых изменяется при регулировании скорости привода. Таким образом, в систему непосредственного управления встроены функции промышленного программируемого контроллера средней производительности, реализующего заданный алгоритм управления технологическим процессом (ПИД-регулятор). Такой подход позволяет отказаться от дополнительных устройств, создавая своеобразные узлы автоматизации, способные самостоятельно управлять как двигателями, так и технологическими переменными (давлением, расходом, подачей и т.д.).

Концепция интеллектуальных распределенных систем управления с каждым годом завоевывает все больше сторонников. Это связано с общей тенденцией переноса все больших вычислительных ресурсов непосредственно к месту, где установлено исполнительное устройство и соответствующий рабочий орган.

В состав системы управления входят следующие изделия.

- Одноплатный универсальный контроллер (в формате E2).
- Пульт оперативного управления с двухстрочным жидкокристаллическим дисплеем и клавиатуре ft 4x4, выпускаемый как в виде встраиваемого изделия, так и виде автономного изделия, подключаемого к контроллеру плоским кабелем.
- Интерфейсная плата релейного ввода/вывода дискретных сигналов «сухим» контактом (8 входов и 8 выходов).
- Интерфейсная плата ввода/вывода импульсных сигналов, обеспечивающая сопряжение с 2-, 3-, 4-, 6-фазными датчиками положения, а также ввод реперных сигналов идентификации начальной позиции. Для работы контроллера необходим внешний источник питания 5 В (1А). Встроенная схема мониторинга питания обеспечивает постоянный контроль уровня внешнего питания и формирование запроса на прерывание процессора при раннем обнаружении падения напряжения ниже допустимого уровня.

В качестве центрального процессора выбран один из самых современных и

высокопроизводительных 16-разрядных микроконтроллеров для управления двигателями Inte 18xС196МН, обеспечивающий функции прямого цифрового управления инверторами напряжения в режиме центрированной синусоидальной ШИМ-модуляции на частотах несущей до 20 кГц с возможностями программного регулирования «мертвого времени» для защиты инвертора от сквозного тока в диапазоне от 0 до 125 мкс.

Все выходы ШИМ-генератора могут дополнительно буферизоваться преобразователями уровня TTL-MOS (опция), что позволяет применять в силовой части привода драйверы различных фирм, в том числе с изолированными источниками питания ключей. Допускается подключение внешних источников питания для преобразователей уровня с диапазоном напряжений от 5 до 24 В. Предусмотрены программно-аппаратные средства безопасной инициализации при подаче напряжения питания на контроллер или при сбросе системы, исключающие случайное формирование отпирающих управляющих сигналов на драйверы силовых ключей в течение переходных процессов в цепях питания.

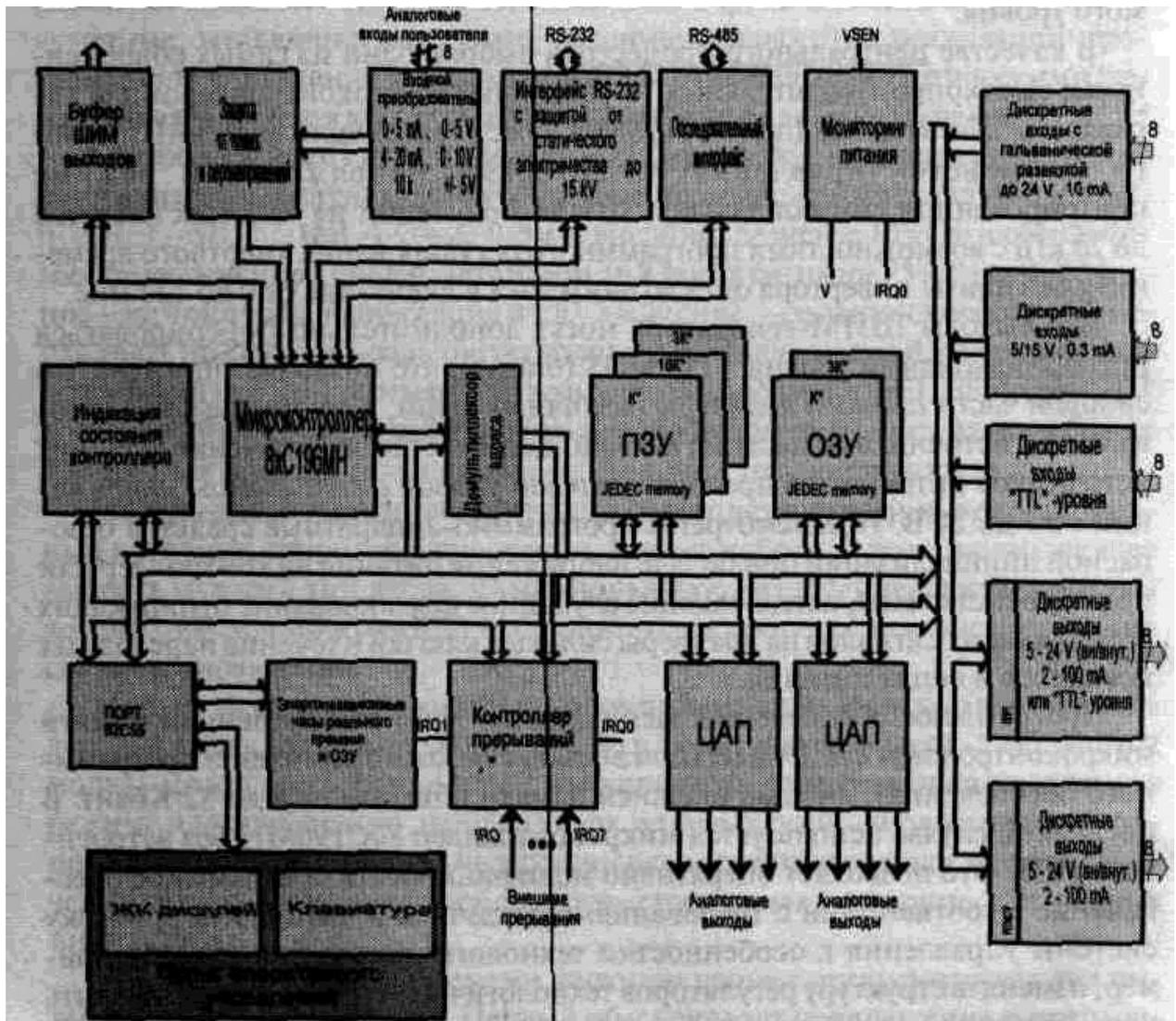


РИС. 17. БЛОК-СХЕМА УНИВЕРСАЛЬНОГО КОНТРОЛЛЕРА

Программное обеспечение располагается либо во внутренней памяти микроконтроллера 87С196МН (при поставке базового комплекта программного обеспечения), либо во внешней памяти объемом от 8 до 32 Кбайт. В последнем случае исполняется микроконтроллер 80С196МН без встроенного ПЗУ. Это позволяет оперативно модифицировать программное обеспечение в соответствии с требованиями заказчика и выполнять привязку системы управления к особенностям технологического процесса, например, изменять структуру регуляторов технологических переменных, вводить новые алгоритмы адаптивного управления и т.д.

Имеется также встроенная защита программного обеспечения от несанкционированного доступа.

Модуль памяти комплектуется различными типами внешней оперативной памяти:

- Статическим ОЗУ объемом от 8 до 32 Кбайт.
- Энергонезависимым ОЗУ объемом от 8 до 32 Кбайт с гарантированных сроком хранения данных до 10 лет.

Статическое ОЗУ используется исключительно для временного хранения промежуточных результатов, а также для ведения оперативного протокола с целью наблюдения за текущими значениями регулируемых переменных в приводе (мощность, ток и т.д.), а также за параметрами технологического процесса (давление в магистрали, расход, концентрация и т.д.).

Энергонезависимой ОЗУ используется для хранения параметров привода, преобразователя, Заводских уставок, а также заданных циклограмм управления технологическими переменными в функции времени (часовые, суточные, недельные, месячные, годовые циклы). Для реализации программного управления координатами привода или технологическими переменными в состав контроллера включены *часы реального времени*.

Контроллер имеет Унифицированный интерфейс с различными источниками аналоговых сигналов в большинстве промышленных стандартов: 0-5В, 0-10В, +/-5В, +/-10В, 4-20мА, 0-5мА, внешний резистор 10 кОм и обеспечивает подключение в общей сложности до 8 таких сигналов (5 из них внутренние - датчики тока, напряжения и т.д., 3 внешние, — задание скорости, обратная связь по скорости, Задание технологической переменной, обратная связь по технологической переменной, возмущающее воздействие). Точность АЦП составит 10 разрядов, время одного преобразования 10 мкс, максимальная абсолютная погрешность не более +/-3 единиц младшего разряда.

Модуль ввода аналоговых сигналов имеет встроенный высокостабильный источник опорного напряжения, а также наборное поле, позволяющее выбрать нужный тип входа. Функциональное назначение каждого из аналоговых входов программно настраивается в процессе конфигурирования системы управления.

Модуль вывода аналоговых сигналов позволяет выводить во вне до 8 аналоговых сигналов с разрешением 1/256 в диапазоне 0-5 В или -5,+5 В. Этот модуль может быть использован для управления шаговыми двигателями в

режиме электрического дробления шага (до 4-х осей одновременно от одного контроллера) или для индикации скорости привода или текущих значений технологических переменных на стрелочных или других аналоговых индикаторах.

Для сопряжения с системами верхнего уровня предусмотрены два интерфейса RS-232 и RS-485. Первый обеспечивает подключение к преобразователю любого компьютера или промконтроллера, в том числе «на ходу».

Эта возможность позволяет выполнить настройку преобразователя или диагностику оборудования с использованием портативных компьютеров в качестве переносных пультов оперативного управления. Второй позволяет подключить к одному каналу связи до 32-х преобразователей частоты и организовать не только управление преобразователями в реальном времени от системы верхнего уровня, но и сбор информации о состоянии привода и локально-контролируемого технологического оборудования.

Последовательные каналы связи дополнительно могут использоваться для подключения удаленных датчиков технологических переменных: температуры, уровня жидкости и т.д.

Контроллер имеет мощную встроенную систему ввода/вывода дискретных сигналов, что позволяет подключать кнопочные станции, командо-аппараты, датчики состояния технологического оборудования и т.д., а также выводить управляющие воздействия дискретного типа и информационные сигналы о состоянии привода.

Входные дискретные сигналы могут быть трех типов: ТТЛ-уровня, 15В и 24 В. Для входных сигналов напряжением 15 В предусмотрена дополнительная защита от помех триггерами Шмитта. Дискретные входы 24 В имеют встроенную гальваническую развязку. Выходные дискретные сигналы могут быть нескольких типов: ТТЛ, с регулируемым выходным напряжением в диапазоне от 5 до 24 В за счет подсоединения внешнего источника питания.

Если по условиям эксплуатации необходимо использовать релейные входы или выходы, то к контроллеру подключается дополнительная интерфейсная плата на 8 входов, 8 выходов.

Система прерываний микроконтроллера 8xС196МН расширена с помощью дополнительной БИС контроллера прерываний, что позволяет вводить по прерываниям ряд дискретных сигналов с датчиков, например, теплового перегрева инвертора, двигателя и т.д.

Возможности системы управления позволяют вводить (и выводить) также импульсные сигналы, например с импульсных датчиков положения, причем с автоматической идентификацией процессором скорости двигателя в широком диапазоне (вплоть до 10000 об/мин.). Измерение скорости производится с автоматическим программным переключением диапазонов с целью получения максимальной точности в каждом диапазоне. Это позволяет применять частотно-регулируемые асинхронные привода в точных замкнутых системах управления. Имеется также ряд дополнительных возможностей по организации

синхронной работы нескольких приводов, по организации управления группой шаговых двигателей (до четырех осей) с автоматическим контролем выпадения двигателей из синхронизма.

Набор программного обеспечения зависит от типа привода. Комплект программного обеспечения для асинхронного привода реализует функции автоматического регулирования выходного напряжения в функции текущей выходной частоты, задатчика интенсивности, блока ограничения резонансных частот, выбора заданного типа торможения, технологического регулятора и т.д. Разумеется реализованы и функции защиты двигателя и преобразователя: максимально-токовая, время-токовая, от сквозных токов, от обрыва и перекося фаз, от недопустимого отклонения напряжения сети и другие.

В состав программного обеспечения входят драйверы дисплея и клавиатуры пульта оперативного управления, драйверы последовательных каналов связи, программе -монитор и другое системное программное обеспечение.

Выводы

- Рассматриваемая в качестве примера версия системы управления преобразователями частоты открывает богатейшие возможности по автоматизированной настройке параметров преобразователя в процессе пуско-наладочных работ, по автоматической адаптации режима работы привода к условиям эксплуатации, по решению различных задач оптимального управления. Она сочетает в себе возможности качественного прямого цифрового управления инвертором с возможностями управления теологическими переменными.

- Пользователь вместе с преобразователем получает и встроенный промконтроллер средней производительности, приобретая возможность создавать узлы локальной автоматизации, которые затем, при необходимости, можно легко подключить по каналу связи к системе управления более высокого уровня, с целью как удалённого управления, так и наблюдения за регулируемыми переменными и состоянием привода.

- Унифицированные интерфейсы позволяют использовать одну и ту же систему управление для приводов различных типов: постоянного тока, синхронных, шаговых, асинхронных и вентильных.

- Управляющие воздействия можно вводить через аналоговые входы, дискретные входы; или по одному из каналов последовательной связи. Гибкая система настройки конфигурации системы управления позволяет выполнять программную перенастройку функционального назначения входов системы управления оптимальным образом.

- Русифицированный дисплей пульта оперативного управления позволяет создавать дружественное программное обеспечение с интуитивно понятной мнемоникой. Возможно конфигурирование системы управления, настройка и местное управление с использованием портативного компьютера, подключенного к системе по последовательному каналу связи.

- При переходе к массовому производству изделий с встроенными системами управления аппаратная часть унифицированного контроллера может быть легко минимизирована в соответствии с требованиями конкретной задачи.

- Использование перспективных систем встроенного управления преобразователями частоты насосов холодного и горячего водоснабжения жилых и промышленных зданий позволяет экономить до 50% электроэнергии и до 15% воды. При этом срок окупаемости преобразователя не превышает одного года. В условиях начавшейся в России коммунальной реформы, актуальность работ в этом направлении не вызывает сомнений.

Лекция 12.

Мультипроцессорные системы управления. Транспьютеры - элементная база мультипроцессорных систем

Основные особенности транспьютеров

Концепция параллелизма давно привлекала внимание специалистов своими потенциальными возможностями повышения производительности и надежности вычислительных систем. В нашей стране с 60-х годов выполнялись теоретические, экспериментальные и промышленные разработки в этом направлении. Именно с подобными системами связывается в настоящее время перспектива дальнейшего наращивания производительности. Исторически первой промышленной разработкой, ориентированной на массово-параллельные системы стали транспьютеры.

Транспьютер - это микрокомпьютер с собственной внутренней памятью и линками (каналами) для соединения с другими транспьютерами. Термин "транспьютер", произошедший в результате объединения слов "транзистор" и "компьютер", отражает основную область его применения - массово-параллельные вычислительные системы, в которых он играет роль базового вычислительного элемента. Некоторые специалисты понимают термин "транспьютер" как название конкретного продукта фирмы Inmos, другие трактуют его как обобщенное наименование микропроцессоров со встроенными межпроцессорными интерфейсами. Используется также термин "транспьютер-подобный микропроцессор", чтобы, с одной стороны, подчеркнуть, что речь идет не о продукте фирмы Inmos, а, с другой стороны, указать, что микропроцессор имеет встроенные линки для образования параллельных систем. Вполне возможно, что стремительное развитие микроэлектроники не позволит термину "транспьютер" устояться, и он будет поглощен более общим - микропроцессор, так как отличительный признак транспьютера - встроенные межпроцессорные интерфейсы - появятся в том или ином виде у всех микропроцессоров.

Первый транспьютер - T414 - был представлен фирмой Inmos, Inc. (Бристоль, Великобритания) в 1983 г. Его основные характеристики:

- разрядность - 32 бит;

- объем внутренней памяти - 2 Кбайта;
- число коммуникационных каналов (линков)- 4;
- скорость обмена по линку - 5, 10, 20 Мбит/с;
- тактовая частота - внешняя - 5МГц, внутренняя - 15 МГц;
- производительность - 10 MIPS.

Широкую доступность и известность транспьютеры получили с 1985 г. Были выпущены модификации с большим объемом памяти (4 Кбайт) и более высокой тактовой частотой - семейство Т-4: Т424, Т425 - 20, 25 и 30 МГц, 16-разрядные модификации - семейство Т-2: Т212, Т222, транспьютеры со встроенным устройством выполнения операций с плавающей точкой - семейство Т-8: Т800, Т801, Т805. Производительность этих микропроцессоров достигает 30 MIPS и 4,3 MFLOPS. Выпускается ряд периферийных устройств транспьютерных семейств, к числу которых относятся микросхемы: М212 - контроллер НЖМД стандарта ST506, G412 - графический RGB-контроллер, С004 - программируемые 32-канальные коммутаторы и др.

Высокая степень "функциональной самостоятельности" транспьютера, простота интеграции и наличие периферийных устройств позволяют в короткие сроки создавать системы на их основе. Коммуникационные каналы транспьютера могут осуществлять обмен данными одновременно с вычислениями, практически не снижая производительности процессора. Благодаря этому качеству транспьютеров системы на их основе обладают хорошей масштабируемостью и высоким значением показателя эффективности - производительность/стоимость.

Архитектура и структура транспьютеров фирмы Inmos

Архитектура семейств Т-2, Т-4, Т-8

Транспьютеры относятся к классу RISC- процессоров. Система команд транспьютеров ориентирована на поддержку языка высокого уровня -ОККАМ (ОССАМ). Это язык параллельного программирования, позволяющий задавать параллельные вычисления в соответствии с моделью взаимодействующих последовательных процессов (CSP). Программа на языке ОККАМ представляет собой совокупность асинхронных совместно протекающих взаимодействующих процессов, выполняющихся асинхронно и параллельно. Под процессом понимается ход исполнения программного кода некоторой программы или фрагмента программы. Взаимодействие между процессами реализуется путем обмена данными по принципу "рандеву". В транспьютере данная модель параллельных вычислений поддерживается благодаря наличию аппаратно-реализованного диспетчера, обеспечивающего выполнение параллельных процессов в режиме квантования времени. Причем количество одновременно выполняемых процессов не ограничено.

Мультипроцессорная система на базе транспьютеров представляет собой совокупность транспьютеров, соединенных линиями связи (непосредственно или через коммутатор).

Существенно упростить программирование мультипроцессорных ВС на базе транспьютеров позволяет тот факт, что одинаковая модель параллельных вычислений поддерживается как внутри отдельного транспьютера, так и в рамках мультитранспьютерной системы в целом. Благодаря этому программа, разрабатываемая для мультипроцессорной системы, может быть создана и отлажена на одном единственном процессоре, а затем перенесена на сеть транспьютеров без существенных преобразований. Исключение составляет ограничение числа каналов связи процесса с процессами, протекающими на другом транспьютере. Это ограничение преодолено в транспьютере T-9000.

Обобщенная структура транспьютера T800 приведена на рис. 76. На этом же рисунке показаны архитектурные регистры процессора, управляющие структуры процессов и распределение их рабочих областей памяти.

В зависимости от модели, транспьютер состоит из 32- или 16-разрядного центрального процессора, интерфейса внешней памяти, 2 или 4 двунаправленных каналов - линков, программируемого блока событий (Event), таймера, внутреннего ОЗУ емкостью 2 или 4 Кбайт, блока режимов работы линков, блока системных функций. Некоторые модели могут содержать 64-разрядное устройство операций с плавающей точкой и (или) интерфейсные схемы внешних устройств, такие как НЖМД, ЭЛТ, сетевой адаптер. (Контроллеры внешних устройств обычно выполняются на кристалле вместо двух линков).

Центральный процессор

32-разрядный центральный процессор транспьютера работает на тактовой частоте до 30 МГц, формируемой из опорной внешней тактовой частоты 5 МГц внутренней схемой умножения. Внутренняя структура центрального процессора показана на рис. 77.

Аппаратный планировщик (диспетчер) центрального процессора организует одновременное (в режиме разделения времени) выполнение нескольких процессов. Каждому процессу в момент его создания в адресном пространстве транспьютера выделяется выровненная по границе слова рабочая область памяти и устанавливается приоритет. В транспьютере реализована двухуровневая схема приоритетов: 0 - высокий приоритет процесса, 1 - низкий приоритет. Объединение адреса рабочей области и приоритета образует дескриптор процесса.

Центральный процессор содержит два регистра-таймера, для низко- и высокоприоритетных процессов. Высокоприоритетный регистр инкриминируется каждую 1 мкс, низкоприоритетный - каждые 64 мкс.

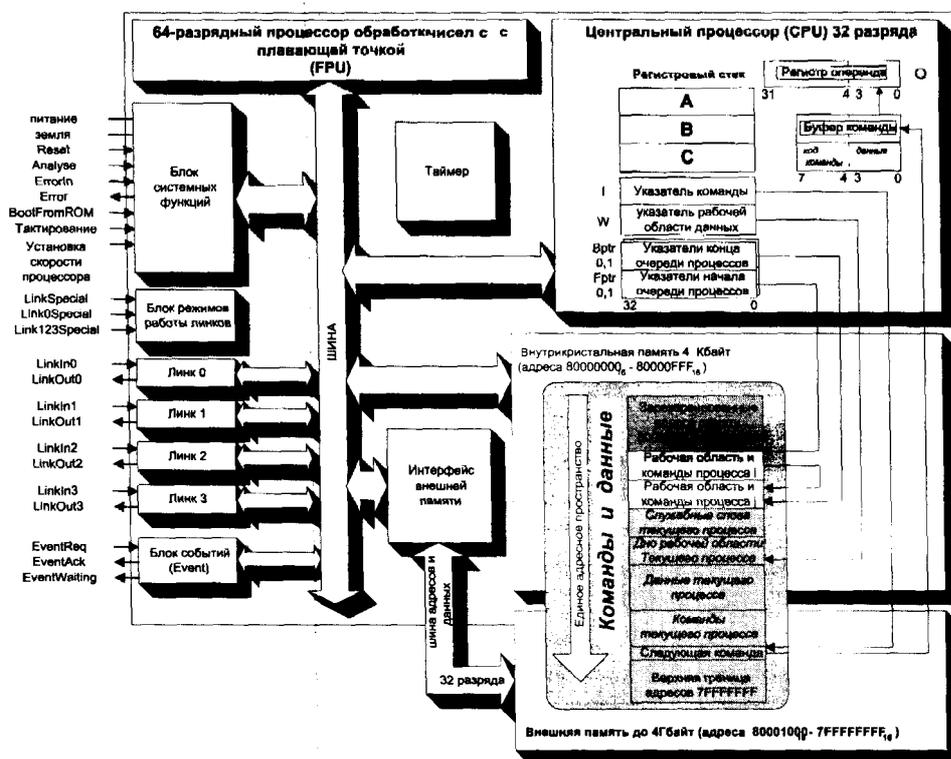


РИС. 76. СТРУКТУРА ТРАНСПЬЮТЕРА T800

Регистр W транспьютера указывает на рабочую область процесса, а в регистре указателе команды находится адрес следующей выполняемой команды.

Регистры общего назначения A, B, C, образуют регистровый стек ss память с дисциплиной доступа FIFO. АЛУ выполняет логические и арифметические операции над операндами, содержащимися в регистровом стеке. Операнды помещаются в стек через его вершину - регистр A. При загрузке данных в A его содержимое переносится в B, содержимое B в C, а содержимое последнего теряется. Результаты операций также формируются в регистре A, при этом содержимое регистра C "выталкивается" в регистр B и становится неопределенным.

Система команд транспьютера

Все команды транспьютера однобайтовые и выполняются за один такт процессора. Структура команды показана на рис. 76. Старшие 4 разряда - код команды, младшие 4 разряда используются для формирования операнда команды или кода команды в регистре операнда - O.

Операции, реализуемые командами транспьютера, подразделяются на первичные и вторичные. 13 наиболее употребительных первичных операций, такие, например, как сложение с константой со значением в диапазоне 0 -15, безусловный переход, загрузка (запись) слова и т.п. Еще 3 команды (мнемокоды rfix, nfix, org) используются для реализации вторичных операций. Код вторичной операции формируется командами rfix, nfix и org в регистре O, а операнды содержатся в регистрах.

Выполнение команд

Команды выбираются из памяти транспьютера и помещаются в буфер команд центрального процессора. За один такт выбирается 2 команды в T-414 и 4 команды в T-800.

Выполнение большинства команд осуществляется в три этапа:

1. Операнд команды помещается в младшие 4 разряда регистра операнда- O;
2. Выполняется операция, задаваемая кодом команды. Содержимое регистра O интерпретируется как операнд.
3. Очищается регистр операнда - O.

Исключение составляют команды rfix, nfix и org.

Команда rfix на этапе 2 осуществляет сдвиг содержимого O на 4 разряда влево и не очищает регистр O на этапе 3. Команда nfix выполняется так же, как и rfix, но перед сдвигом O помещает туда арифметическое дополнение его содержимого.

Поскольку после выполнения команд rfix и nfix содержимое O не очищается, последовательность команд rfix и nfix позволяет формировать в регистре O операнды с разрядностью до 32. Требуемая последовательность команд минимальной длины формируется компилятором.

Команда org интерпретирует содержимое O как код вторичной операции. В этом случае операнды содержатся в регистрах процессора. Код вторичной операции, таким образом, задается как операнд первичной операции org и может иметь до 2^{32} различных значений. В существующих на сегодняшний день моделях транспьютеров используется около сотни вторичных операций.

Использование сопроцессора

В случае наличия у транспьютера встроенного сопроцессора, выполняющего операции с плавающей точкой, распознавание команд и передача их на выполнение осуществляются аппаратно в центральном процессоре, там же выполняется вычисление адреса операндов и их загрузка в регистры FPU.

Модуль операций с плавающей точкой, как показано на рис. 77, состоит из двух блоков: блока мантиссы и блока порядка со своим набором регистров (два регистровых стека). Все команды с плавающей точкой подразделяются на два класса: полностью самостоятельные команды, никак не влияющие на состояние центрального процессора, и команды, пересылающие результаты выполнения операции в центральный процессор.

После передачи команды на исполнение в сопроцессор центральный процессор продолжает выполнение потока команд, если это команда относится к первому классу, или ожидает получения результата, если это команда второго класса.

Распределение памяти транспьютера

Транспьютер может адресовать до 2^{32} байт (4 Гбайт). Особенностью транспьютера является начало адресов в области отрицательных значений. Младший адрес в дополнительном коде соответствует значению (80000000...).

Структурно (конструктивно) вся оперативная память подразделяется на внутрикристальную и внешнюю. Объем внутрикристальной памяти составляет 4 (2) Кбайт (в зависимости от модели транспьютера). Архитектурно (с точки зрения программиста) вся память транспьютера, как внешняя, так и внутрикристальная, равнодоступна и имеет единую адресацию. Внутрикристальной памяти соответствуют младшие адреса, а внешней памяти - старшие.

Поскольку внутрикристальная память имеет меньшее время доступа (один такт процессора), в нее обычно помещаются программистом часто используемые данные и подпрограммы. Пример распределения памяти показан на рис 76.

Несколько слов в младших адресах памяти используется для специальных целей: для слов состояния аппаратных каналов - линков, блока Event, регистров таймера, в качестве рабочих областей памяти диспетчера.

Диспетчеризация процессов

Каждый из процессов в конкретный момент времени может быть выполняемым, активным (готовым к выполнению в очереди соответствующего приоритета), ожидающим сигнала таймера (в очереди к таймеру соответствующего приоритета), ожидающим ввода-вывода.

Для выполняемого процесса регистр I содержит адрес следующей команды, регистр W содержит его дескриптор.

Для организации очереди процессов на выполнение (очередь активных процессов) в диспетчере используются пары регистров FptrO, BptrO, FptrI, BptrI, указывающие на начало и конец очереди высоко- и низкоприоритетных процессов соответственно. В качестве элементов очередей используются рабочие области процессов, содержащие помимо данных процесса, управляющую информацию, требующуюся для сохранения и восстановления состояния процессов на этапе диспетчеризации (в том числе ссылку на рабочую область следующего процесса в очереди).

Низкоприоритетные процессы выполняются в течение не более чем 32 периодов низкоприоритетного таймера, после чего процесс (если он еще способен к выполнению) помещается в конец очереди активных низкоприоритетных процессов, а диспетчер планирует выполнение следующего процесса. Сначала просматривается очередь высокоприоритетных процессов, затем - низкоприоритетных.

Высокоприоритетный процесс не прерывается и выполняется до тех пор, пока это возможно (до завершения или до момента ожидания ввода-вывода, сигнала таймера или сигнала внешнего события, выдаваемого блоком Event). Если высокоприоритетный процесс становится активным (например, получает сигнал таймера) в момент выполнения процесса с низким приоритетом, последний прерывается, его состояние сохраняется в области зарезервированных адресов памяти транспьютера и иницируется выполнение прерванного высокоприоритетного процесса. После его выполнения

возобновляется прерванный низкоприоритетный процесс.

Ввод-вывод

Ввод-вывод для процессов в транспьютере организуется одинаковым образом как в случае использования аппаратных линков (внешний ввод-вывод), так и в случае использования виртуального линка (обмен между процессами одного транспьютера). В обмене всегда участвуют только два процесса: один вводит данные, другой выводит.

В момент выполнения команды описания канала и после завершения каждого обмена слово состояния канала (ССК) инициализируется константой Minint. Процесс, первым выполняющий команду обмена, приостанавливается и проверяется содержимое канала. Если оно Minint, то в ССК заносится дескриптор процесса, а в рабочую область процесса помещаются значение W и адрес передаваемых данных, после чего управление передается диспетчеру для планирования выполнения других процессов.

Если содержимое ССК не Minint, значит данный процесс "пришел" на "рандеву" вторым. В этом случае содержимое ССК - дескриптор процесса, используя который находятся данные в случае ввода или буфер для данных в случае вывода.

Обмен по линкам осуществляется аналогично, с той разницей, что ССК имеют фиксированные адреса в памяти 80000000 - 8000001C,g, и в процессе обмена данные не пересылаются между областями памяти, а передаются по линии связи в стартстопном режиме с квитированием байтов.

При выполнении ввода-вывода от центрального процессора требуется лишь инициализация обмена. По команде ввода-вывода вся требуемая информация об операции обмена - адрес и длина данных - помещается во внутренние регистры линка, после чего центральный процессор освобождается для дальнейших вычислений, а линк транспьютера самостоятельно управляет обменом данными.

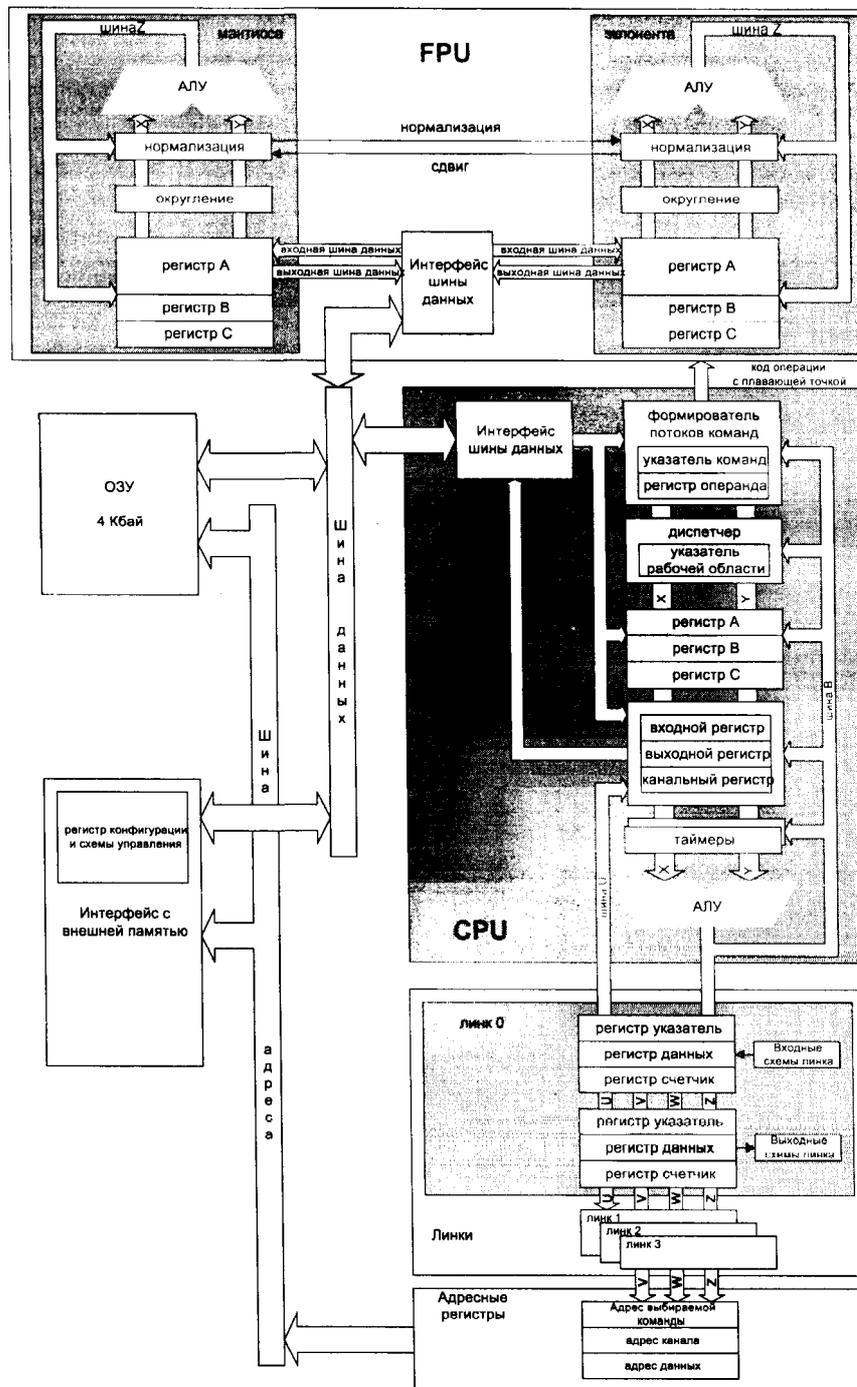


РИС. 77. ВНУТРЕННЯЯ СТРУКТУРА ФУНКЦИОНАЛЬНЫХ БЛОКОВ ТРАНСПЬЮТЕРА

Передача данных по линку

Все семейства транспьютеров Т-2 (Т-212, Т-222, Т-225), Т-4 (Т-414, Т-400, Т-425), Т-8 (Т-800, Т-801, Т-805) используют один и тот же протокол передачи данных по линкам. Взаимодействие между транспьютерами осуществляется посредством обмена сообщениями, состоящими из последовательности байтов. Данные передаются по одному проводу из пары проводов. По другому проводу пары передаются подтверждения приема каждого байта.

Байт передается в обрамлении служебных битов, начиная со стартового бита, далее следует управляющий бит, а затем 8 информационных битов и столбый бит. Таким образом, на 8 информационных битов приходится 3 управляющих,

поддерживающих протокол передачи. На рис. 78. показан пример передачи по линку.

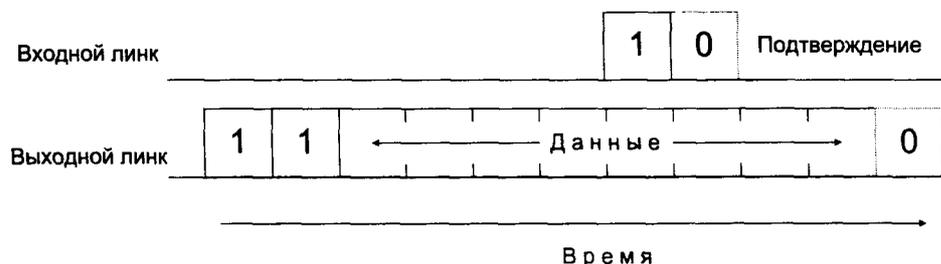


РИС. 78. ПЕРЕДАЧА ДАННЫХ ПО ЛИНКУ ТРАНСПЬЮТЕРА

После передачи байта отправитель ждет подтверждения, которое состоит из стартового и управляющего бита. Причем если при передаче информационного байта управляющий бит равен 1, то в подтверждении он равен 0. При этом байты данных и подтверждения для переданных в противоположном направлении байтов данных передаются по одному проводнику. Подтверждения имеют приоритет перед байтами данных. Можно установить режим, при котором подтверждения начинают передаваться сразу после получения стартового и управляющего битов, что приводит к передаче байтов без задержки между ними.

Если один из обменивающихся через линк процессов не готов к приему данных, байты накапливаются в регистре данных линка. При заполнении регистра подтверждение после получения очередного байта не выдается, и передающий линк приостанавливает передачу до приема данных процессом и освобождения регистра линка.

При инициации передачи первого байта сообщения существует задержка на настройку каналов прямого доступа в передающем и приемном транспьютерах, что обуславливает разную скорость при передаче коротких (от единиц до нескольких десятков байтов) и длинных (несколько тысяч байтов) сообщений. В первом случае скорость составляет 2-4 Мбит/с, во втором - максимально возможную 10-20 Мбит/с, в зависимости от настройки передающего и принимающего транспьютеров.

Передача по линку асинхронна, приемный транспьютер не чувствителен к фазе принимаемых сигналов. Единственное, что требуется - точный кварцевый генератор 5 МГц для задания тактовых сигналов транспьютеров.

Блок режимов работы линков позволяет задавать скорость передачи по линкам транспьютера (5, 10, 20 Мбит/с), причем для нулевого линка скорость устанавливается независимо от остальных. Установка осуществляется подачей соответствующих уровней на входы LinkSpecial, LinkOSpecial, Link123Special.

Обмены по линку программируются отдельно в передающем и принимающем транспьютерах. В передающем транспьютере программируется команда передачи по линку заданного числа байтов. В принимающем транспьютере программируется команда приема из линка задаваемого числа байтов. Реализация обмена происходит, если программы в передающем и принимающем транспьютерах выходят на команды передачи и приема. Если один транспью-

тер вышел на соответствующую команду, то он ждет, пока другой выйдет на свою команду обмена. Неправильно запрограммированный обмен ведет к бесконечному ожиданию. Кроме того, бесконечное ожидание возможно, если заданы разные значения количества передаваемых и принимаемых байтов.

Ожидание сигнала от блока событий

Блок событий (Event) выполняет преобразование внешних логических уровней на входе блока в байтовое сообщение со значением 0 или 1, передаваемое по внутреннему каналу. Поэтому программно блок Event воспринимается как специальный канал, отличием которого от обычного канала является то, что из этого канала можно только "читать". ССК для блока Event имеет фиксированный адрес 80000020. Ожидать ввода из канала Event может одновременно только один процесс в транспьютере.

Обычно Event используется для регистрации внешних прерываний в транспьютерной системе. Сигнал прерывания подается на вход транспьютера EventReq. Подтверждение приема сигнала прерывания (чтение из канала Event) выдается в виде уровня 1 по линии EventAck.

Ожидание сигнала от таймера

Программно таймер воспринимается как канал только для чтения, выдающий либо содержимое регистра таймера с соответствующим приоритетом, либо сигнал наступления ожидаемого момента времени.

Все процессы, ожидающие наступления некоторого момента времени (истечения временного интервала) помещаются в очередь к таймеру, соответствующему их приоритету. Очередь процессов отсортирована в порядке наступления ожидаемого момента времени и организована с помощью ссылок между служебными словами в рабочей области процессов. Адрес первого процесса в очереди соответствующего приоритета содержится в служебных словах транспьютера (TPtrLocO, TPtrLocI), в младших адресах памяти. При достижении ожидаемого момента времени диспетчеру передается дескриптор соответствующего процесса для его перевода в конец очереди активных процессов.

Инициализация системы после включения питания

Транспьютер и его ОЗУ построены по КМОП-технологии и не сохраняют своего состояния после отключения питания. Поэтому после включения питания для начала функционирования в транспьютер необходимо загрузить некоторый минимум программного обеспечения. Транспьютер может быть загружен как из внешнего ПЗУ, так и из любого линка. Для указания режима начальной загрузки используется вывод BootFromROM. Если уровень BootFromROM равен 1, то управление передается по адресу FFFFFFFE, по которому обычно содержится команда безусловного перехода на программу начальной загрузки.

Если BootFromROM равен 0, то производится загрузка из линка. После включения питания транспьютер переходит в состояние ожидания приема данных по линкам. Первый байт, принятый по любому из линков, управляет

дальнейшим режимом работы транспьютера. Если его значение больше 1, то он интерпретируется как длина кода программы, принимаемой следом за ним. Принимаемые данные записываются в память транспьютера начиная с адреса MemStart, с него же и начинается выполнение программы. Как правило, первой загружается программа начальной загрузки транспьютера, которая обеспечивает загрузку остального программного обеспечения.

Если первый принятый управляющий байт равен 0 или 1, то транспьютер переходит в режим управления памятью.

Управляющий байт, равный 0, заставляет транспьютер интерпретировать следующие 4 байта как адрес памяти, по которому в память будет записано слово, поступившее как следующие 4 байта. После этого транспьютер снова переходит в режим управления памятью, из которого его может вывести лишь прием управляющего байта со значением, большим 1.

Если управляющий байт равен 1, следующие 4 байта задают адрес слова памяти, которое будет считано и передано как 4 байта по тому же линку, из которого поступили в противоположном направлении предыдущие 4 байта. После этого транспьютер остается в режиме управления памятью.

Режим управления памятью используется обычно для целей отладки.

Управление системой

Появление уровня 1 на входе Analyze переводит транспьютер, работающий по программе, в режим управления памятью. Как уже было сказано выше, в этом режиме может быть считано и изменено состояние памяти транспьютера.

Обработка ошибок

Программные ошибки, такие как арифметическое переполнение, деление на 0, выход за границы массива, вызывают установку в транспьютере флага error и появление сигнала на выходе Error. Флаг режима обработки ошибок транспьютера - HaltOnError - позволяет определять поведение транспьютера в случае ошибки (установки флага error): если был задан режим остановки (HaltOnError 1), то в случае возникновения ошибки устанавливается единица на выходе Error и транспьютер останавливается; если была выполнена установка HaltOnError=0, то в случае ошибки устанавливается в единицу флаг error, однако транспьютер продолжает работать.

В мультипроцессорных системах выводы Analyze, Reset, Error, Errorin всех транспьютеров обычно соединены согласно схеме, представленной на рис. 79. При такой схеме соединения появление сигнала Error на выводе любого транспьютера переводит всю систему в режим управления памятью. Соответствующие программы в хост-машине позволят определить состояние системы и пути преодоления ошибочной ситуации.

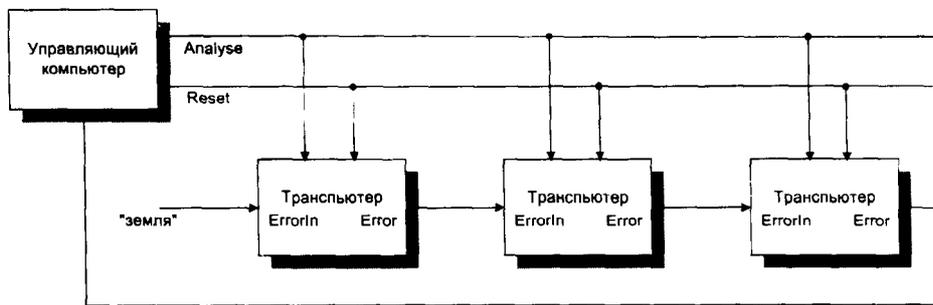


РИС. 79. СХЕМА СОЕДИНЕНИЯ ЛИНИЙ СИСТЕМОГО СЕРВИСА В МУЛЬТИТРАНСПЬЮТЕРНОЙ СИСТЕМЕ

Лекция 13.

Транспьютер T9000

Архитектурные и структурные особенности

На момент своего появления транспьютеры семейства T-8 были самыми быстродействующими 32-разрядными микропроцессорами. Попытка фирмы Inmos сохранить лидерство перед американскими производителями микропроцессоров воплотилась в разработку транспьютера T9000. Его основные технические характеристики:

- производительность - 200 MIPS, 25 MFLOPS;
- объем внутрикристальной памяти - 16 Кбайт;
- число коммуникационных каналов - 4;
- скорость обмена по линку - 100 Мбит/с.

Основными особенностями архитектуры данного транспьютера являются аппаратная поддержка механизма виртуальных каналов и аппаратный группировщик команд, повышающий загрузку параллельно функционирующих устройств процессора.

Однако фирма Inmos не смогла выдержать обещанные сроки поставки T-9000, а также достичь объявленной производительности. Поэтому, хотя образцы T-9000, работающие на пониженной, по сравнению с объявленной, тактовой частоте были выпущены, коммерческого успеха они не имели.

Виртуальные линки

Механизм виртуальных линков позволяет вести по одному физическому линку обмен между произвольным числом пар процессов, протекающих в разных транспьютерах.

Управляет обменом встроенный в T-9000 процессор виртуального канала (VCP). Сообщение, передаваемое от процесса-отправителя к процессу-получателю, VCP делит на пакеты, каждый из которых содержит 32 байта данных (последний пакет - от 1 до 32 байт), заголовок пакета и концевик (дам последнего пакета - признак конца сообщения, для остальных - признак конца

пакета). При получении пакета, VSP в принимающем транспьютере передает подтверждение в виде пустого пакета содержащего только заголовок и признак конца пакета. VSP, используя информацию, содержащуюся в заголовке пакета, осуществляет маршрутизацию пакетов и сшивку сообщения. Таким образом, обмен данными для процессов выглядит так же, как и в случае транспьютеров прежних поколений, что способствует преемственности программного обеспечения.

"Прозрачность" для процессов маршрутизации сообщений в сети транспьютеров T9000 полностью устраняет различие между обменом в рамках одного транспьютера и обменом в транспьютерной сети. Это свойство существенно упрощает разработку программы для мультитранспьютерной системы и повышает ее эффективность, поскольку не требуется дополнительных расходов на организацию маршрутизации и коммутации.

В целях увеличения числа физических связей транспьютера T9000 разработан программируемый коммутатор С 104, осуществляющий передачу сообщения с любого из 32 входов на любой из 32 выходов в соответствии с их заголовком.

Для возможности использования в системе T9000 совместно с транспьютерами предшествующих поколений разработана микросхема С 100, выполняющая согласование электрических характеристик и преобразование формата передаваемых по линкам данных.

Группировщик команд

В T-9000 полностью сохранена система команд предыдущих поколений транспьютеров. Увеличение производительности достигается за счет одновременного исполнения группы, в которую входит до 8 команд.

В T-9000 реализован аппаратный группировщик команд. Образование групп команд преследует цель достижения высокой загрузки устройств процессора.

Процессор за один такт извлекает из памяти 4 команды. В силу того, что некоторые команды требуют для исполнения более чем 1 такт, в процессоре может быть накоплено количество команд, достаточное для формирования 5 групп по 8 команд каждая, что соответствует полной загрузке устройств процессора.

Транспьютероподобные микропроцессоры серии "Квант"

Примером отечественной разработки транспьютероподобных процессоров являются микропроцессоры серии "Квант". Это семейство 32-разрядных микропроцессоров с оригинальной архитектурой, сочетающей RISC-подход с методикой длинного командного слова. Семейство характеризуется высокой степенью внутреннего параллелизма процессов обработки, конвейерным выполнением команд, Гарвардской архитектурой памяти, наличием последовательных коммуникационных каналов -линков. Были выпущены две модификации микропроцессора:

- "Квант-10" по технологии 2,5 мкм на трех полузаказных матричных

кристаллах 1537 ХМ2, спроектированных в НИИ "Квант" и изготовленных в НИИ точной технологии, г. Зеленоград;

- "Квант-20" по технологии 1,5 мкм на одном кристалле типа U1700 фирмы ZMD (ФРГ, Дрезден), спроектированном в НИИ "Квант" и изготовленном фирмой ZMD. Структура микропроцессора показана на рис. 80. Микропроцессор работает с внешними отдельными кэш-памятями команд и данных и может использовать сопроцессор. Память данных микропроцессора содержит только данные, тогда как память команд может содержать как команды, так и данные (такая архитектура памяти получила название модифицированной Гарвардской).

Устройство управления

Процессор выбирает из памяти команд 32-разрядные команды и помещает их сначала в первый, а затем во второй регистры команд, далее декодирует команды и формирует управляющие сигналы для всех функциональных блоков процессора. Блок управления прерываниями обеспечивает приоритетную схему обработки 10 типов внешних и внутренних прерываний. Блок ПДП прямого доступа в память позволяет выполнять вычисления одновременно с обменом данными по 4 коммуникационным каналам. Блок защиты данных предназначен для защиты фрагментов памяти команд и данных, выделяемых под операционную систему.

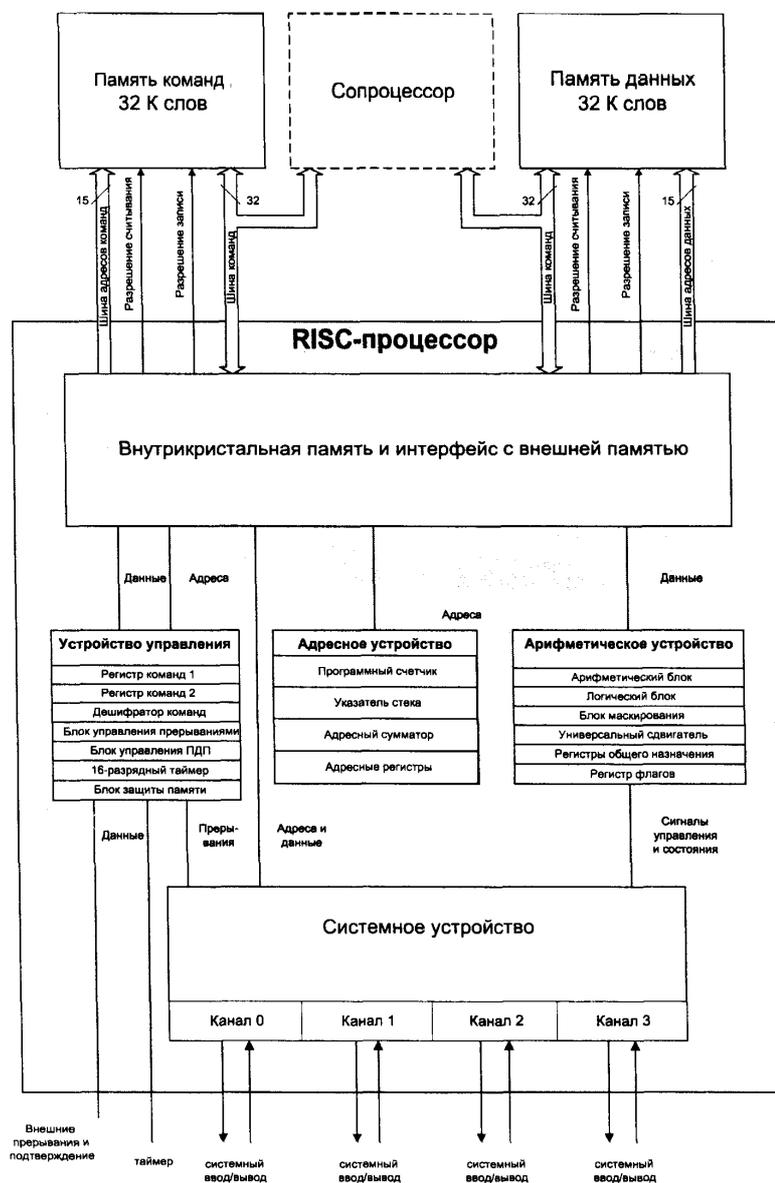


Рис. 80. Структура микропроцессора "Квант"

Адресное устройство

В процессоре реализована концепция разнесенной (decoupled) архитектуры, в соответствии с которой все вычисления адресов выполняются отдельным адресным устройством, что предоставляет возможность одновременной обработки данных в арифметическом устройстве и вычисления адресов в адресном устройстве.

Адресное устройство содержит в программном счетчике адрес следующей команды, поддерживает в памяти стек адресов возврата из подпрограмм (обработчиков прерываний), выполняет все вычисления адреса в 16-разрядном адресном сумматоре. Файл адресных регистров (AR) содержит 8 16-разрядных AR для МП "Квант-10" и 4 16-разрядных AR для МП "Квант-20".

Арифметическое устройство

Арифметический блок способен выполнять 16 логических и 14 арифметических операции, в том числе байтовые, пошаговое умножение 32-

разрядного множимого на два разряда множителя. Логический блок может выполнять логические операции параллельно с арифметическими. Все операции выполняются над 32-разрядными операндами за один такт.

Универсальный сдвигатель способен выполнять за один такт логический, арифметический или циклический сдвиг 32-разрядного слова влево или вправо на 0 - 31 разряд, а также осуществлять циклические сдвиги внутри байтов, тетрад, пар. Схема маскирования арифметического устройства позволяет маскировать результаты любой операции содержимым одного из регистров общего назначения. Файл регистров общего назначения (РОН) используется для хранения операндов, результатов, масок, адресов данных. В регистр флагов заносятся признаки по результатам выполнения операций в арифметическом устройстве.

Системное устройство

Системное устройство обеспечивает связь с 4 аналогичными процессорами по независимым каналам. Обмен осуществляется побитно блоками слов. В начале каждого блока задается количество передаваемых слов и адрес в памяти, в который будет записываться сообщение. В процессе передачи осуществляется контроль по четности для каждого передаваемого байта данных. В случае ошибки при передаче вырабатывается соответствующее прерывание.

Регистр защиты памяти позволяет запрещать запись в любой блок памяти длиной 4 Кслов.

Конвейер процессора

В процессоре реализован трехстадийный конвейер выполнения команд. На первой стадии осуществляется выборка команды из памяти команд, на второй стадии производится формирование адреса данных для последующего обращения в память и модификация регистров адреса, на третьей стадии выполняются ввод-вывод данных из памяти по предварительно вычисленному адресу и операции арифметики. Действия на каждом этапе выполняются за один такт, что позволяет при обеспечении высокой степени загруженности конвейера выполнять команды в среднем за один такт.

Система команд

Команды процессора подразделяются на простые и комплексные. Первые выполняют одно действие, тогда как вторые задают трехадресную арифметическую операцию над данными в регистрах одновременно с операцией обмена данными с памятью и (или) модификацией адресных регистров.

При обращении к памяти используются следующие виды адресации:

базовая по содержимому АР, базовая по содержимому РОНа, автоинкрементная или автодекрементная адресация по любому АР, базово-индексная адресация по двум АР. В МП "Квант-20" добавлена базово-индексная адресация с 8-разрядным смещением, задаваемым в поле команды.

Для упрощения устройства управления и обеспечения большей гибкости в программах в микропроцессоре реализована следующая схема выполнения

условных и безусловных переходов. В случае условного перехода специальная команда проверяет соответствие флага признаков результата арифметической операции одному из 16 возможных условий перехода. Если имеет место соответствие, то следующая команда не выполняется, а как бы подменяется пустой операцией (NOP). Причем проверка условия осуществляется на фоне выполнения арифметических операций.

Чтобы не нарушить работу конвейера (избежать пропуска конвейерных циклов), безусловный переход выполняется по принципу "отложенного перехода". В конвейере сначала отрабатывается команда следующая за командой перехода, а затем выполняется переход, так же выполняется и обращение к подпрограммам.

Конвейеризация внутренних процессов в осуществляется во времени выполнения операций в различных функциональных устройствах микропроцессора позволяют выполнять до четырех команд за один такт.

Производительность микропроцессора.

Гибкая система команд микропроцессор способствует его эффективному применению как на задачах счётного характера, так и на задачах логической и символьной обработки. Коммуникационные возможности процессора позволяют строить на его базе масштабируемые системы с MPP-архитектурой.

Уникальная архитектура процессора серии "Квант" позволила обеспечить лучшее значение производительности по сравнению с транспьютерами фирмы Inmos. Производительность микропроцессора "Квант-10" с тактовой частотой 4 МГц соответствует производительности транспьютера T-800 с частотой 20 МГц. Производительность микропроцессоров "Квант-10" и "Квант-20" может быть оценена как 12-15 Моп/с и 25-30 Моп/с.

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



**Методические указания по организации самостоятельной
работы и задания по дисциплине**

**Б1.В.09.09 ОСНОВЫ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ В
МАШИНОСТРОЕНИИ**

Специальность -

21.05.04 Горное дело

Специализация -

Мехатроника и робототехника промышленных комплексов

Автор: Зубов В.В., к.т.н., Адас В.Е.

Одобен на заседании кафедры

Эксплуатации горного оборудования

(название кафедры)

Зав. кафедрой

Симисин Д.И.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

(Дата)

Рассмотрен методической комиссией
факультета

Горно-механического

(название факультета)

Председатель

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

ВЕДЕНИЕ 3

ВИДЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА.....	4
МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ КО ВСЕМ ВИДАМ ТЕКУЩЕГО КОНТРОЛЯ.....	4
Повторение материала лекций и самостоятельное изучение курса	4
Подготовка к практическим занятиям и лабораторным работам.....	5
Подготовка и написание контрольной работы	6
Подготовка к выполнению и написанию курсовой работы (проекта)	7
МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	8
Подготовка к зачёту.....	8
Подготовка к экзамену	8

Самостоятельная работа студентов — это разнообразные виды деятельности студентов, осуществляемые под руководством, но без непосредственного участия преподавателя в аудиторное и/или внеаудиторное время.

Это особая форма обучения по заданиям преподавателя, выполнение которых требует активной мыслительной, поисково-исследовательской и аналитической деятельности.

Методологическую основу самостоятельной работы студентов составляет деятельностный подход, когда цели обучения ориентированы на формирование умений решать типовые и нетиповые задачи, то есть на реальные ситуации, где студентам надо проявить знание конкретной дисциплины, использовать внутрпредметные и межпредметные связи.

Цель самостоятельной работы - закрепление знаний, полученных на аудиторных занятиях, формирование способности принимать на себя ответственность, решать проблему, находить конструктивные выходы из сложных ситуаций, развивать творческие способности, приобретение навыка организовывать своё время

Кроме того, самостоятельная работа направлена на обучение студента осмысленно и самостоятельно работать сначала с учебным материалом, затем с научной информацией, заложить основы самоорганизации и самовоспитания с тем, чтобы привить умение в дальнейшем непрерывно повышать свой профессиональный уровень.

Самостоятельная работа реализует следующие задачи:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирование практических (общеучебных и профессиональных) умений и навыков;
- развитие исследовательских умений;
- получение навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная образовательной программой, учебным планом и учебно-методическими материалами, раскрывающими и конкретизирующими их содержание, осуществляется студентами инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует студентам источники и учебно-методические пособия для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т. п.

ВИДЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА

Основные формы организации самостоятельной работы студентов определяются следующими параметрами:

- содержание учебной дисциплины;
- уровень образования и степень подготовленности студентов;
- необходимость упорядочения нагрузки студентов при самостоятельной работе.

В соответствии с реализацией рабочей программы дисциплины в рамках самостоятельной работы студенту необходимо выполнить следующие виды работ:

для подготовки ко всем видам текущего контроля:

- повторение материала лекций;
- самостоятельное изучение курса;
- подготовка к практическим занятиям и лабораторным работам;
- подготовка к контрольной работе, написание контрольной работы;
- выполнение и написание курсовой работы (проекта);

для подготовки ко всем видам промежуточной аттестации:

- подготовка к зачёту;
- подготовка к экзамену.

Особенностью организации самостоятельной работы студентов является необходимость не только подготовиться к сдаче зачета /экзамена, но и собрать, обобщить, систематизировать, проанализировать информацию по темам дисциплины.

Технология организации самостоятельной работы студентов включает использование информационных и материально-технических ресурсов образовательного учреждения.

Самостоятельная работа может осуществляться индивидуально или группами студентов как вне, так и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

В качестве форм и методов контроля внеаудиторной самостоятельной работы студентов могут быть использованы обмен информационными файлами, семинарские занятия, тестирование, опрос, доклад, реферат, самоотчеты, контрольные работы, защита контрольных и курсовых работ (проектов), защита зачётных работ в виде доклада с презентацией и др.

Текущий контроль результатов внеаудиторной самостоятельной работы студентов осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине.

Промежуточный контроль результатов внеаудиторной самостоятельной работы студентов осуществляется в пределах времени, отведенного для сдачи экзамена / зачёта.

В методических указаниях по каждому виду контроля представлены материалы для самостоятельной работы и рекомендации по организации отдельных её видов.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ КО ВСЕМ ВИДАМ ТЕКУЩЕГО КОНТРОЛЯ

Повторение материала лекций и самостоятельное изучение курса

Лекционный материал по дисциплине излагается в виде устных лекций преподавателя во время аудиторных занятий. Самостоятельная работа студента во время лекционных аудиторных занятий заключается в ведении записей (конспекта лекций).

Конспект лекций, выполняемый во время аудиторных занятий, дополняется студентом при самостоятельном внеаудиторном изучении некоторых тем курса. Самостоятельное изучение тем курса осуществляется на основе списка основной и дополнительной литературы к дисциплине.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины приведён в рабочей программе дисциплины.

Рекомендуемые задания для самостоятельной внеаудиторной работы студента, направленные на повторение материала лекций и самостоятельное изучение тем курса: *для овладения знаниями:*

- конспектирование текста;
- чтение основной и дополнительной литературы;
- составление плана текста;
- работа со словарями, справочниками и нормативными документами;
- *просмотр обучающих видеозаписей. для закрепления и систематизации знаний:*
- работа с конспектом лекций;
- повторная работа над учебным материалом;
- составление таблиц для систематизации учебного материала;
- изучение нормативных материалов;
- составление плана и тезисов ответа на вопросы для самопроверки;
- ответы на вопросы для самопроверки;
- составление библиографических списков по изучаемым темам. *для*

формирования навыков и умений:

- выполнение рисунков, схем, эскизов оборудования;
- рефлексивный анализ профессиональных умений.

Тематический план изучения дисциплины и содержание учебной дисциплины приведены в рабочей программе дисциплины.

Вопросы для самопроверки приведены учебной литературе по дисциплине или могут быть предложены преподавателем на лекционных аудиторных занятиях после изучения каждой темы.

Подготовка к практическим занятиям и лабораторным работам

Практические занятия по дисциплине выступают средством формирования у студентов системы интегрированных умений и навыков, необходимых для освоения профессиональных компетенций, а также умений определять, разрабатывать и применять оптимальные методы решения профессиональных задач.

На практических занятиях происходит закрепление теоретических знаний, полученных в ходе лекций, осваиваются методики и алгоритмы решения типовых задач по образцу и вариантных задач, разбираются примеры применения теоретических знаний для практического использования, выполняются доклады с презентацией по определенным учебно-практическим, учебно-исследовательским или научным темам с последующим их обсуждением.

Рекомендуемые задания для самостоятельной внеаудиторной работы студента, направленные на подготовку к практическим занятиям: *для овладения знаниями:*

- чтение основной и дополнительной литературы;
- работа со словарями, справочниками и нормативными документами;
- *просмотр обучающих видеозаписей. для закрепления и систематизации знаний:*
- работа с конспектом лекций;
- ответы на вопросы для самопроверки;
- подготовка публичных выступлений;
- составление библиографических списков по изучаемым темам. *для*

формирования навыков и умений:

- решение задач по образцу и вариативных задач;
- выполнение рисунков, схем, эскизов оборудования;

- рефлексивный анализ профессиональных умений.

Тематический план изучения дисциплины и содержание учебной дисциплины приведены в рабочей программе дисциплины.

Лабораторные занятия по дисциплине выступают средством формирования у студентов навыков работы с использованием лабораторного оборудования, планирования и выполнения экспериментов, оформления отчётной документации по выполнению лабораторных работ.

Рекомендуемые задания для самостоятельной внеаудиторной работы студента, направленные на подготовку к лабораторным занятиям: *для овладения знаниями:*

- изучение методик работы с использованием различных видов и типов лабораторного оборудования;
- изучение правил безопасной эксплуатации лабораторного оборудования;
- работа со словарями, справочниками и нормативными документами. *для закрепления и систематизации знаний:*

- составление плана проведения эксперимента;
- составление отчётной документации по результатам экспериментирования;
- аналитическая обработка результатов экспериментов. *для формирования навыков и умений:*

- выполнение рисунков, схем, эскизов оборудования;
- оформление отчётной документации по выполнению лабораторных работ.

Подготовка и написание контрольной работы

Контрольная работа - индивидуальная деятельность обучающегося по концентрированному выражению накопленного знания, обеспечивает возможность одновременной работы всем обучающимся за фиксированное время по однотипным заданиям, что позволяет преподавателю оценить всех обучающихся. Контрольная работа является средством проверки умений применять полученные знания для решения задач определенного типа по теме или разделу.

Рекомендуемые задания для самостоятельной внеаудиторной работы студента, направленные на подготовку к контрольной работе: *для овладения знаниями:*

- чтение основной и дополнительной литературы;
- работа со словарями, справочниками и нормативными документами. *для закрепления и систематизации знаний:*

- работа с конспектом лекций;
- ответы на вопросы для самопроверки. *для формирования навыков и умений:*
- решение задач по образцу и вариативных задач;
- выполнение рисунков, схем, эскизов оборудования;
- оформление отчётной документации по выполнению контрольной работы.

Контрольная работа может быть выполнена в виде доклада с презентацией.

Доклад с презентацией — это публичное выступление по представлению полученных результатов знаний по определенной учебно-практической, учебно-исследовательской или научной теме.

При подготовке доклада с презентацией обучающийся должен продемонстрировать умение самостоятельного изучения отдельных вопросов, структурирования основных положений рассматриваемых проблем, публичного выступления, позиционирования себя перед коллективом, навыки работы с библиографическими источниками и оформления научных текстов.

В ходе подготовки к докладу с презентацией обучающемуся необходимо:

- выбрать тему и определить цель выступления;
- осуществить сбор материала к выступлению;
- организовать работу с источниками;
- во время изучения источников следует записывать вопросы, возникающие по мере ознакомления, ключевые слова, мысли, суждения; представлять наглядные примеры из практики;
- сформулировать возможные вопросы по теме доклада, подготовить тезисы ответов на них;
- обработать материал и представить его в виде законченного доклада и презентации.

При выполнении контрольной работы в виде доклада с презентацией самостоятельная работа студента включает в себя: *для овладения знаниями:*

- чтение основное и дополнительной литературы по заданной теме доклада;
- составление плана доклада;
- работа со словарями, справочниками и нормативными документами;
- просмотр обучающих видеозаписей по теме доклада

для закрепления и систематизации знаний:

- составление плана и тезисов презентации по теме доклада;
- составление презентации;
- составление библиографического списка по теме доклада;
- подготовка к публичному выступлению;
- составление возможных вопросов по теме доклада и ответов на них. *для*

формирования навыков и умений:

- публичное выступление;
- выполнение рисунков, схем, эскизов оборудования;
- рефлексивный анализ профессиональных умений.

Варианты контрольных работ и темы докладов приведены в комплекте оценочных средств дисциплины.

Подготовка к выполнению и написанию курсовой работы (проекта)

Курсовая работа (проект) - форма контроля для демонстрации обучающимся умений работать с объектами изучения, критическими источниками, справочной и энциклопедической литературой, логично и грамотно излагать собственные умозаключения и выводы, обосновывать и строить априорную модель изучаемого объекта или процесса, создавать содержательную презентацию выполненной работы.

При выполнении и защите курсовой работы (проекта) оценивается умение самостоятельной работы с объектами изучения, справочной литературой, логично и грамотно излагать собственные умозаключения и выводы, обосновывать выбранную технологическую схему и принятый тип и количество оборудования, создавать содержательную презентацию выполненной работы (пояснительную записку и графический материал).

Рекомендуемые задания для самостоятельной внеаудиторной работы студента, направленные на подготовку к курсовой работе (проекту): *для овладения знаниями:*

- чтение основной и дополнительной литературы;
- работа со словарями, справочниками и нормативными документами;
- составление плана выполнения курсовой работы (проекта);
- составление списка использованных источников.

для закрепления и систематизации знаний:

- работа учебно-методическими материалами по выполнению курсовой работы (проекта);
- изучение основных методик расчёта технологических схем, выбора и расчёта оборудования;
- подготовка тезисов ответов на вопросы по тематике курсовой работы (проекта).

для формирования навыков и умений:

- решение задач по образцу и вариативных задач;
- выполнение рисунков, схем, компоновочных чертежей;
- оформление текстовой и графической документации.

Тематика курсовых работ (проектов) приведены в комплекте оценочных средств дисциплины.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Подготовка к зачёту

Зачёт по дисциплине может быть проведён в виде теста или включать в себя защиту контрольной работы (доклад с презентацией).

Тест — это система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.

При самостоятельной подготовке к зачёту, проводимому в виде теста, студенту необходимо:

- проработать информационный материал (конспект лекций, учебное пособие, учебник) по дисциплине; проконсультироваться с преподавателем по вопросу выбора дополнительной учебной литературы;

проведения теста: выяснить условия
количество вопросов в тесте,

продолжительность выполнения теста, систему оценки результатов и т. д.;

- приступая к работе с тестом, нужно внимательно и до конца прочитать вопрос и предлагаемые варианты ответов, выбрать правильные (их может быть несколько), на отдельном листке ответов вписать цифру вопроса и буквы, соответствующие правильным ответам.

В процессе выполнения теста рекомендуется применять несколько подходов в решении заданий. Такая стратегия позволяет максимально гибко оперировать методами решения, находя каждый раз оптимальный вариант. Не нужно тратить слишком много времени на трудный вопрос, а сразу переходить к другим тестовым заданиям, к трудному вопросу можно обратиться в конце. Необходимо оставить время для проверки ответов, чтобы избежать механических ошибок.

Зачёт также может проходить в виде защиты контрольной работы (доклад с презентацией). Методические рекомендации по подготовке и выполнению доклада с презентацией приведены в п. «Подготовка и написание контрольной работы».

Подготовка к экзамену

Промежуточная аттестация по итогам освоения дисциплины проводится в форме экзамена.

Билет на экзамен включает в себя теоретические вопросы и практик ориентированные задания.

Теоретический вопрос - индивидуальная деятельность обучающегося по концентрированному выражению накопленного знания, обеспечивает возможность

одновременной работы всем обучающимся за фиксированное время по однотипным заданиям, что позволяет преподавателю оценить всех обучающихся.

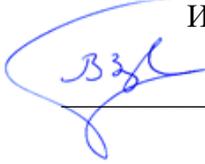
Практико-ориентированное задание - средство проверки умений применять полученные знания для решения задач определенного типа по определенной теме.

При самостоятельной подготовке к экзамену студенту необходимо:

- получить перечень теоретических вопросов к экзамену;
- проработать пройденный материал (конспект лекций, учебное пособие, учебник) по дисциплине, при необходимости изучить дополнительные источники;
- составить планы и тезисы ответов на вопросы;
- проработать все типы практико-ориентированных заданий;
- составить алгоритм решения основных типов задач;
- выяснить условия проведения экзамена: количество теоретических вопросов и практико-ориентированных заданий в экзаменационном билете, продолжительность и форму проведения экзамена (устный или письменный), систему оценки результатов и т. д.;
- приступая к работе с экзаменационным билетом, нужно внимательно прочитать теоретические вопросы и условия практико-ориентированного задания;
- при условии проведения устного экзамена составить план и тезисы ответов на теоретические вопросы, кратко изложить ход решения практикоориентированного задания;
- при условии проведения письменного экзамена дать полные письменные ответы на теоретические вопросы; изложить ход решения практико-ориентированного задания с численным расчётом искомых величин.

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»


И. о. проректора по учебно-методической работе
В. В. Зубов



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Б1.В.09.10 ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ МЕХАТРОННЫХ И РОБОТОТЕХНИЧЕСКИХ УСТРОЙСТВ

Специальность -

21.05.04 Горное дело

Специализация -

Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Информатики

(название кафедры)

Зав. кафедрой


(подпись)

Дружинин А.В.

(Фамилия И.О.)

Протокол № 1 от 19.09.2024

(Дата)

Рассмотрен методической комиссией факультета

Горно-механического

(название факультета)

Председатель


(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. АБСТРАКТНЫЕ ТИПЫ ДАННЫХ И ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	6
2. ПРИМЕРЫ АБСТРАКТНЫХ ТИПОВ ДАННЫХ.....	9
3. ОБЗОР СРЕДСТВ ОПИСАНИЯ КЛАССОВ	11
3.1. Классы и члены	11
3.2. Конструктор и деструктор.....	12
3.3. Встроенные (inline) функции.....	14
3.4. Класс vector	17
3.5. Перегрузка операторов	19
3.6. Указатель this.....	21
3.7. Дружественные функции-операции.....	22
3.8. Класс очередь que. Шаблон класса	22
3.9. Наследование. Производные классы	24
4. ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ.....	27
4.1. Необходимость обработки исключений.....	27
4.2. Общие принципы обработки исключительных ситуаций в C++.....	29
4.3. Возбуждение ситуации	30
4.4. Обработка исключений	32
5. РЕАЛИЗАЦИЯ ПРИМЕРОВ	34
5.1. Класс полином	34
5.2. Класс таблица	36
6. ЗАДАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ.....	39
ЛИТЕРАТУРА	41

ВВЕДЕНИЕ

Язык С++ предоставляет гибкие и эффективные средства определения новых типов. Используя определения новых типов, отвечающих понятиям проблемной области (приложениям), программист может разбивать разрабатываемую программу на части, легко поддающиеся контролю.

Новый тип создается пользователем для определения понятия, которому среди встроенных типов нет соответствия. Гораздо легче понять программу, в которой создаются типы, отвечающие понятиям проблемной области. Хорошо определенные типы делают программу ясной и короткой. Кроме того, компилятор может обнаружить недопустимое использование данных.

Такой метод построения программ называют абстракцией данных. Информация о типах содержится в объектах, тип которых определяется пользователем. Программирование с применением объектов называют объектно-ориентированным [1,2].

Язык С++ является языком объектно-ориентированного программирования (ООП). Автор языка создавал его с целью поддержки абстракции данных и объектно-ориентированного программирования. Родоначальниками языков такого типа являются языки Smalltalk и Simula 67.

Объектно-ориентированный язык – это язык программирования, на котором программа задается описанием поведения совокупности взаимосвязанных объектов. Объектно-ориентированное программирование имеет дело с объектами.

Объектно-ориентированные языки включают следующие основные черты: инкапсуляция данных, полиморфизм, наследование.

Объекты могут включать закрытые (private) данные и правила их обработки, доступные только объекту, защищенные (protected), доступные объекту и его наследникам, а также общие (public) данные и правила, которые доступны объектам и модулям в других частях программы. Важной чертой ООП является наследование, то есть возможность создавать иерархическую последовательность объектов от более общих к более специфическим объектам.

Методические указания содержат четыре основных раздела.

В первом разделе обсуждается взаимосвязь абстрактных типов данных и объектно-ориентированного программирования.

Во втором разделе приводится список примеров, в которых вводится понятие, являющееся абстракцией.

В третьем разделе дается краткий обзор средств языка C++, необходимых для реализации абстрактного понятия, причем вводимые средства показываются на конкретных примерах. Для определения нового (пользовательского) типа вводится понятие класс. С понятием класс непосредственно связаны понятия конструктор и деструктор. В этом же разделе рассматриваются простейшие средства для реализации производных классов.

Четвертый раздел посвящен обработке исключительных ситуаций, возникающих в процессе работы программ.

Пятый раздел представляет собой реализацию конкретных абстрактных понятий, а именно, реализацию полинома и таблицы.

В шестом разделе представлены задания для практических занятий.

1. АБСТРАКТНЫЕ ТИПЫ ДАННЫХ И ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

В языках высокого уровня можно выделить следующие категории средств определения данных:

- встроенные типы данных (integer, real, character, boolean);
- средства структурирования составных объектов (array, record, union);
- средства определения новых типов данных – это абстракция данных;
- средства отображения иерархии типов данных.

Понятие абстрактного типа [1] пришло в программирование из математики. Абстрактный объект – это то, что является предметом математического рассуждения, состоящего из определений, допущений (или постулатов) и утверждений, выводимых из определений и допущений по общепонятным правилам логического вывода (например, хорошо знакомые математикам: множества, функции, последовательности и рекурсивные структуры).

Организация данных в языках программирования базируется на понятии *типа*. К особенностям понятия типа следует отнести следующие:

- 1) тип определяет класс значений, которые могут принимать переменная или выражение;
- 2) каждое значение принадлежит одному и только одному типу;
- 3) тип значения константы, переменной или выражения можно вывести либо из контекста, либо из вида самого операнда, не обращаясь к значениям, вычисляемым во время работы программы;
- 4) каждой операции соответствует некоторый фиксированный тип ее операндов и некоторый фиксированный тип результата;

5) для каждого типа свойства значений и элементарных операций над значениями задаются с помощью аксиом;

б) при работе с языком высокого уровня знание типа позволяет обнаруживать в программе бессмысленные конструкции и решать вопрос о методе представления данных и преобразования их в вычислительной машине.

Абстракция – это инструмент познавательной деятельности человека, позволяющий лучше отразить суть дела и приводящий к абстрактным понятиям. Поэтому имеет смысл говорить о средствах абстракции в языках программирования, которые сами являются средством понимания и построения алгоритмов и обмена мыслями и результатами между программистами.

Абстрактный тип данных в языках программирования – это определение некоторого понятия в виде класса объектов с некоторыми свойствами и операциями. Такое определение оформляется как специальная синтаксическая конструкция, которая называется *классом* в языке C++.

В определение абстрактного типа данных входят следующие четыре части.

1) Внешность, содержащая имя определяемого типа (понятия), имена операций с указанием типов их аргументов и значений.

2) Абстрактное описание операций и объектов, с которыми они работают, средствами некоторого языка спецификаций.

3) Конкретное описание операций и объектов средствами языка программирования.

4) Описание связи между (2) и (3), объясняющее, в каком смысле часть (3) корректно представляет часть (2).

Внешность – это видимая часть определения, его интерфейс. Отметим, что спецификация определяет то, что важно для пользователя. А для пользо-

вателя существенным является «поведение», то есть «то, что делается», а несущественным – «то, как это делается» [1].

Главное достоинство абстракции через спецификацию состоит в несущественности способа реализации, что позволяет изменять реализацию без внесения изменений в программу.

Основные особенности средства абстракции данных:

1) инкапсуляция описания и представления объектов определяемого типа и описания операций над объектами;

2) защита инкапсулированной информации, так что детали представления не доступны вне определения абстрактного типа.

Средства абстракции данных в языке C++ включают классы, механизм управления доступом, конструкторы и деструкторы, совместное использование операций (перегрузка операций), преобразование типов, полиморфизм, обработку исключительных ситуаций.

Средства отображения иерархии классов непосредственно связаны с объектно-ориентированным программированием, для поддержки которого в языке C++ введены производные классы и виртуальные функции.

2.ПРИМЕРЫ АБСТРАКТНЫХ ТИПОВ ДАННЫХ

Спецификация абстрактных типов данных может быть выполнена на специальном формализованном языке [1]. Приведем примеры абстрактных типов данных, используя неформальный язык.

Пример 1. Опишем понятие стек (stack). Стек представляет собой последовательность элементов некоторого типа, которая характеризуется свойством: "последний вошел – первый вышел" (Last In - First Out) и над которой выполняются следующие операции:

- создать стек;
- положить в стек;
- проверить, пуст ли стек;
- проверить, переполнен ли стек;
- взять элемент, находящийся в вершине стека, удаляя его из стека;
- показать элемент, находящийся в вершине стека, т.е. взять элемент, находящийся в вершине стека, не удаляя его из стека.

Пример 2. Опишем понятие очередь (Queue). Очередь представляет собой последовательность элементов некоторого типа, которая характеризуется свойством: "первый вошел – первый вышел" (First In - First Out) и над которой выполняются следующие операции:

- создать очередь;
- послать в очередь;
- проверить, пуста ли очередь;
- проверить, переполнена ли очередь;
- взять элемент.

Пример 3. Опишем понятие вектор (vector). Вектор представляет собой последовательность из n элементов со следующими операциями:

- создать вектор с заданным количеством элементов;
- добавить вектор;
- вычесть вектор;
- присвоить вектор вектору;
- показать вектор.

Пример 4. Опишем понятие полином (polynom). Полином – это алгебраическое выражение вида

$$y = a_0 \cdot x^n + a_1 \cdot x^{n-1} + \dots + a_{n-1} \cdot x + a_n$$

Это значит, что полином определяется степенью n и последовательностью из $n+1$ коэффициентов. Над полиномом предусматриваются следующие операции:

- создать полином по заданной степени и коэффициентам;
- прибавить заданный полином;
- вычесть заданный полином;
- взять наибольшую степень полинома;
- взять коэффициент полинома при заданной степени;
- вычислить полином по заданному значению аргумента;
- выдать полином.

Пример 5. Опишем понятие таблицы. Таблица – это последовательность элементов заданной структуры со следующими операциями:

- создать таблицу с заданным количеством элементов;
- выбрать элементы таблицы по условию;
- сортировать таблицу по различным полям;
- показать таблицу.

3. ОБЗОР СРЕДСТВ ОПИСАНИЯ КЛАССОВ

В определении нового типа главное – отделить внешность, которая существенна для его правильного использования, от конкретной реализации, связанной с форматом данных для хранения объекта типа и операций над объектом. Доступ к таким типам ограничен заданным множеством операций (функциями доступа).

Средством определения нового типа в C++ является понятие класса.

3.1. Классы и члены

Класс – это определяемый пользователем тип. Класс специфицирует данные, необходимые для представления объекта этого типа, и множество операций для работы с объектами.

Определение класса имеет две части: закрытую часть (`private`), содержащую информацию для разработчика, и открытую часть (`public`), являющуюся интерфейсом для пользователя.

Доступ к объектам класса ограничивается множеством функций, которые описываются в интерфейсной части класса. Такие функции называются *функциями – членами*. Пользователь может обращаться только к этим функциям.

Опишем тип стек (`stack`) (пример 1 в разделе 2). Стек будем располагать в динамической памяти. Поэтому элемент стека – это структура, состоящая из информационной части и ссылки на следующий элемент.

```
class STACK
{ private:
    struct sp
```

```

        { int inf;
          struct sp *next;
        };
    sp *s;
    public:
CREATE();           // создать стек
int top();         // показать элемент
push(int x);      // добавить элемент
int pop();        // взять элемент
int empty();      // проверить, пуст ли стек
int full();       // проверить, переполнен ли стек
};

```

Для того чтобы пользоваться объектом стек, его нужно описать:

```
STACK st;
```

Объект `st` создаётся и инициализируется функцией-членом `CREATE()`, специально описанной для этой цели. При этом пользователь должен не забыть вызвать функцию `CREATE()`:

```
st.CREATE();
```

Но в языке C++ есть более удобный подход, а именно, определяется специальная функция, предназначенная для инициализации объекта. Эта функции называется *конструктором*.

3.2. Конструктор и деструктор

Конструктор – это предписание, как создавать значение данного типа. Конструктор распознается по тому, что имеет то же имя, что и сам класс. Поэтому в примере роль операции создать стек `CREATE()` выполняет специальная функция `STACK()`. Функция вызывается автоматически при объявлении объекта типа `STACK`.

Определяемый пользователем тип всегда имеет конструктор, так как необходимы выделение памяти и инициализация объектов.

Для обеспечения уничтожения объектов используется специальная функция класса, называемая *деструктором*. Когда закончена работа с объектом, автоматически вызывается деструктор.

Деструктор имеет такое же имя, как и класс, только перед ним ставится знак тильды (~).

Для класса STACK деструктором является ~STACK().

Опишем объект стек с использованием конструктора и деструктора, а также выполним реализацию функций-членов.

Вариант 1.

```
class STACK
{ private:
    struct sp
    {   int inf;
        struct sp *next;
    };
    sp *s;
public:
    STACK() { s=0;}
    ~ STACK()
    { while (s!=0) delete s;}
    int top()
    { return s->inf;
    }

    push(int x)
    { sp *p;
      p=new sp;  p->inf=x; p->next=s;
      s=p;
    }

    int pop()
    { sp *p;  p=s;
      int r=p->inf; s=p->next;
      delete p;
      return r;
    }
    int empty()
```

```

        { return s==0;
        }
int full()
{ sp *p;
  p=new sp;
  if (p!=0)
    { delete p; return 1; }
  else return 0;
}
};

```

3.3. Встроенные (inline) функции

Функции-члены, реализованные в классе, являются встроенными (inline). Это значит, что после компиляции в объектном коде программы пользователя вместо вызовов функций-членов будут подставлены уже преобразованные тела функций, т.е. произойдёт макроподстановка. Другими словами нет затрат на вызов функции.

Удобно реализацию функций-членов выполнять вне класса. Но в этом случае описатель inline должен быть указан явно. Следует отметить, что описатель inline эффективно использовать для коротких операций (таких, как в нашем случае со стеком).

Для того чтобы вне класса был доступ к членам класса, используется операция :: разрешения области видимости. Операнд в левой части операции :: должен быть именем класса.

Вариант 2.

```

class STACK
{ private:
  struct sp
  { int inf;
    struct sp *next;
  };
  sp *s;

```

```

    public:
    STACK() { s=0;}
    ~STACK()
        { if (s!=0) delete s;}
    int top();
    push(int x);
    int pop();
    int empty();
    int full();
};

inline int STACK :: top()
{ return s->inf;
}

STACK :: push(int x)
{ sp *p;
  p=new sp;  p->inf=x; p->next=s;
  s=p;
}

inline int STACK :: pop()
{ sp *p;
  p=s;
  int r=p->inf;s=p->next;
  delete p;
  return r;
}

inline int STACK :: empty()
{ return s == 0;
};

inline int STACK :: full()
{ sp *p;
  p=new sp;
  if (p!=0)
    { delete p; return 1; }
  else return 0;
};

```

Конструктор и деструктор тоже можно определять вне класса.

Следует отметить, что конструктор не может иметь тип возвращаемого значения. Нельзя вызывать функцию–конструктор в явном виде.

Приведем программу, в которой используется класс стек для решения следующей задачи: проверить правильность расстановки круглых скобок в арифметическом выражении.

Алгоритм анализа скобок заключается в следующем:

- если встретилась левая скобка, то она заносится в стек;
- если встретилась правая скобка, то она выталкивает из стека левую скобку в случае, если стек оказался не пуст, иначе устанавливается ошибочная ситуация;
- если по окончании просмотра арифметического выражения стек пуст, то проверка прошла успешно, иначе устанавливается ошибочная ситуация.

Будем предполагать, что класс STACK находится в заголовочном файле stack.h. Описание этого класса соответствует варианту 1 или варианту 2 с той лишь разницей, что информация в стеке будет символьного типа.

Программа:

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <string.h>
#include <stdlib.h>
#include "stack.h"
void main()
{ STACK a;
  char x;
  int n, i, B;
  char *s;
  cout << "\n vvod  str";
  cin >> s;
  n=strlen(s);
  i=0;  B=1;
  while (B && (i<n))
```

```

    {   if (s[i]=='(')   a.push(s[i]);
        if (s[i]==')')
            if (a.empty()) B=0; else x=a.pop();
        i++;
    }
if ((B) && (i==n) && (a.empty()) )   cout <<"\n yes";
else cout <<"\n no";
getch();
}

```

3.4. Класс vector

Конструктор аналогично другим функциям может иметь параметры. Кроме того, класс может иметь несколько конструкторов.

Рассмотрим класс `vector` с точки зрения использования различных конструкторов.

Когда объект создается, бывает необходимо присваивать элементам объекта конкретные значения. Это достигается передачей значений в качестве параметров. В классе `STACK` использовался конструктор без параметра. Конструктор без параметра называется конструктором по умолчанию, конструктор с параметром называется конструктором инициализации. Конструктор с параметром, который имеет тип класса называется копирующим конструктором (*copy-конструктор*). В классе `vector` используются конструкторы с одним параметром, двумя параметрами и *copy-конструктор*.

```

class vector
{ private:
    int *v;   int s;
public:
    vector(int s1)
        { s=s1;   v=new int[s];   }
    vector(int _n,int *n);
    vector(const vector &y);
}

```

```

        ~vector()
        { delete [ ]v; };
void    vw_vect();
void show();
};

vector::vector(int s1,int *n)
{ s=s1;
  v=new  int[s];
  for(int i=0;i<s;i++)
      v[i]=n[i];
};

vector::vector(const vector &y)
{ s=y.s;
  v=new  int[s];
  for(int i=0;i<s;i++)
      v[i]=y.v[i];
}

void vector::vw_vect()
{ cout<<"\nVvod vectora\n";
  for (int i=0;i<s;i++)
      cin>>v[i];
  cout<<"\nVector vveden\n";
};

void vector::show()
{ for (int i=0;i<s;i++)
      cout<<v[i]<<" ";
};

```

Приведём примеры описания объекта класса vector.

Описание: vector a(n);

К моменту этого описания значение n должно быть определено. Автоматически будет вызван конструктор

```
vector(int s1);
```

и вектору a будет выделена память для n значений целого типа.

Описание: `vector d(m,y);`

Автоматически будет вызван конструктор

`vector(int s1,int *n)`

и в этом случае вектору `d` будет выделена память для `m` значений целого типа и скопированы `n` значений из области `y`.

Описание: `vector c(a);`

Автоматически будет вызван *copy-конструктор*

`vector(vector &y)`

и элементам вектора `c` будут присвоены значения элементов вектора `a`.

Приведем фрагменты использования класса `vector`.

```
int n,m;
int y[]={6,3,6,5,9};
cout <<"\n введите n,m " <<" ";
cin >> n >> m;
vector a(n);
a.vv_vect();
a.show();
```

```
vector d(m,y);
d.show();
vector c(a);
a.show();
```

3.5. Перегрузка операторов

Определим в классе `vector` операторы (операции): добавление вектора (символ `+`), вычитание вектора (символ `-`) и присваивание вектора (символ `=`).

```
class vector
{ private:
    int *v;int s;
  public:
```

```

vector(int s1)
    { s=s1;   v=new int[s];
      }
vector(int _n,int *n);
vector( vector &y);
~vector()
    { delete []v; };
void vv_vect();
void show();
vector operator+(vector n);
vector operator-(vector n);
vector operator=(vector n);

};

```

Вне класса определим реализацию введённых операторов.

```

vector vector::operator+(vector n)
{
    vector t(s);
    for (int i=0;i<s;i++) t.v[i]=v[i]+n.v[i];
    return t;
};

```

```

vector vector::operator-(vector n)
{
    vector t(s);
    for (int i=0;i<s;i++) t.v[i]=v[i]-n.v[i];
    return t;
};

```

```

vector vector::operator=(vector n)
{
    if (n.s<s) s=n.s;
    for (int i=0;i<s;i++) v[i]=n.v[i];
    return *this;
};

```

Приведём фрагмент программы, использующий описанные операторы.

```
a=a-d;
cout<<"\n vect a=a-d "<<" " ; a.show();
a=a+d;
cout<<"\n vect a=a+d "<<" " ; a.show();
```

Возможно, что для объекта `vector` естественней было бы использовать вызовы функций, а именно, для операции вычесть `a.minus(d)`, добавить `a.add(d)`.

3.6. Указатель `this`

Каждой функции–члену передается указатель на объект, для которого она вызвана. Таким указателем является служебное слово `this`. Доступ к объекту внутри функции – члена производится с помощью этого указателя `this`.

Служебное слово `this` используется при перегрузке операций.

Операции сложения и вычитания реализуем, используя указатель `this`:

```
vector vector :: operator+(vector n)
{ if (n.s<s) s=n.s;
  for (int i=0;i<s;i++) v[i]=v[i]+n.v[i];
  return *this;
};
vector vector :: operator-(vector n)
{
  if (n.s<s) s=n.s;
  for (int i=0;i<s;i++) v[i]=v[i]-n.v[i];
  return *this;
};
```

3.7. Дружественные функции-операции

Функции-операции могут быть членами класса или дружественными функциями класса. При объявлении дружественной функции должны передаваться два параметра для бинарных операций и один для унарных операций. Дружественными функциями не могут перегружаться операции присваивания `=`, индексирования `[]`, а также `()` и `->`.

Рассмотрим перегрузку операции сложения `+` для класса `vector` с помощью дружественной функции. Описание в классе будет иметь вид:

```
friend vector operator + (vector v1, vector v2);
```

Реализация этой операции вне класса:

```
vector operator + (vector v1, vector v2)
{ int s;
  if (v1.s<v2.s) s=v1.s ; else s=v2.s;
  vector t(s);
  for (int i=0;i<s;i++) t.v[i]=v1.v[i]+v2.v[i];
  return t;
}
```

3.8. Класс очередь `que`. Шаблон класса

Средством реализации полиморфизма в языке C++ является шаблон класса. На примере объекта очередь покажем использование шаблона класса: `template <class T>`.

Это описание указывает на то, что описывается параметризованный тип, имеющий параметром тип `T`.

```
template <class T>
```

```

class QUE
{ private:
  struct link
  { T inf;
    link *next;
  };
  link *start,*end;
  public:
  QUE ();
  ~QUE ();
  void add(T elem);
  T get();
  int empty();
};

```

Приведем реализацию введенных операций.

```

template <class T>
QUE <T>:: QUE ()
{ start=end=0;
};
template <class T>
void QUE <T> :: add(T elem)
{ link *p;
  p=new link;
  p->inf=elem;
  p->next=0;
  if (start==0 && end==0) start=p;
  else end->next=p;
  end=p;
};
template <class T>
T QUE <T> :: get()
{ link *p;
  T buf;
  p=start;
  buf=p->inf;
  start=start->next;
  delete p;
  return buf;
};
template <class T>
QUE <T>::~~ QUE ()

```

```

{ link *p;
  while (start!=0)
    {
      p=start;
      start=start->next;
      delete p;
    };
};

```

Приведем фрагменты программы, использующие класс очередь.

```

QUE <int> ocher;
int x;
scanf ("%d", &x);
ocher.add(x);
printf ("%d\n", ocher.get());
.....
QUE <float> ocher1;
int m;
float y;
scanf ("%f", &y);
ocher1.add(y);
printf ("%f\n", ocher1.get());

```

Аналогично можно описать параметризованный класс стек или класс вектор.

3.9. Наследование. Производные классы

Наследование является важной особенностью объектно-ориентированных языков. Для того чтобы отобразить иерархические связи, выражающие общность между классами, вводится понятие производного класса.

В языке C++ наследуемый класс называют базовым классом, наследующий класс – производным классом. Производный класс наследует свой-

ства базового класса. Поэтому отношение базовый–производный между классами называется наследованием.

Покажем наследование классов на примере объекта очередь. Пусть нам нужно над последовательностью элементов очереди выполнить дополнительно следующую операцию: определить, есть ли среди элементов нулевые. Основные операции над объектом определены в классе QUE. Это операции: добавить элемент

```
void add(T elem);
```

и взять элемент

```
T get();
```

Для простоты опишем класс очередь QUERY без использования шаблона класса.

```
class QUERY
{ private:
  struct link
  { int inf;
    link *next;
  };
  link *start,*end;
public:
  QUERY ();
  ~QUERY ();
  void add(int elem);
  int get();
  int empty();
};
```

Для реализации новой операции можно ввести производный класс, ссылаясь на класс QUERY как базовый. Назовем производный класс QUE_0.

```
class QUE_0 : public QUERY
{ public:
  int def_0 ();
};
```

Вне класса опишем новую операцию:

```

int QUE_0 ::def_0 ()
{ link *p=start;
  while (p!= end) and (p->inf !=0 )
    p=p->next;
  if (p->inf == 0 ) return 1;
  else return 0;
}

```

Когда один класс наследует другой класс, все элементы, определенные как `private` в базовом классе, не имеют доступа в производном классе.

Для того чтобы в классе `QUE_0` был доступ к данным класса `QUERY`, необходимо использовать доступ `protected`. Описание класса `QUERY` будет иметь вид:

```

class QUERY
{ protected:
  struct link
  { int inf;
    link *next;
  };
  link *start,*end;

  public:
  QUERY();
  ~QUERY();
  void add(int elem);
  int get();
  int empty();
};

```

В нашем примере конструктор базового класса представлен без параметра. В этом случае производный класс может не иметь конструктор.

4. ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

4.1. Необходимость обработки исключений

Во время своей работы программа иногда сталкивается с нештатными ситуациями. Например, вызванный метод некоторого объекта может обнаружить у себя внутренние проблемы (неверные значения полей, в результате чего может произойти, например, деление на ноль) или найти ошибки в других объектах или входных данных. Следовательно, необходим некоторый механизм обнаружения и обработки нештатных ситуаций (исключений) в программе.

Когда программа конструируется из отдельных модулей и особенно когда эти модули находятся в независимо разработанных библиотеках, обработка ошибок должна быть разделена на две части:

- 1) генерация информации о возникновении ошибочной ситуации, которая не может быть разрешена локально;
- 2) обработка ошибок, обнаруженных в других местах.

Автор библиотеки может обнаружить ошибки времени выполнения, но, как правило, не знает, что делать в этом случае. Пользователь библиотеки может знать, как поступить в случае возникновения ошибок, но не в состоянии их обнаружить (иначе их поиск не перепоручался бы библиотеке). Для помощи в решении подобных проблем в языке C++ введено понятие *исключения*. Фундаментальная идея состоит в том, что функция (метод), обнаружившая проблему, но не знающая, как ее решать, *возбуждает* (throw) исключение в надежде, что вызвавшая её функция (прямо или косвенно) может решить проблему. В функции, которая хочет решать проблемы данного типа, можно указать, что она *перехватывает* (catch) такие исключения.

Такой стиль обработка ошибок предпочтительнее многих традиционных методов. Рассмотрим альтернативы. При обнаружении проблемы, которая не может быть решена локально, программа может:

- а) прекратить выполнение,
- б) вернуть специальное «ошибочное» значение,
- в) вернуть какое-то допустимое значение и оставить программу в ненормальном состоянии,
- г) вызвать функцию, предназначенную для обработки ошибки.

Вариант а) – прекратить выполнение – это то, что происходит по умолчанию, когда не перехватывается исключение. Библиотечная функция, безусловно завершающая выполнение, не может использоваться в программе, первое требование к которой – надежность.

Вариант б) – вернуть специальное «ошибочное» значение – не всегда выполним, т.к. часто просто нет приемлемого ошибочного значения (например, при возврате целого любое из них может быть приемлемо). Даже когда такой подход применим, он часто неудобен, потому что он вынуждает программиста каждый раз проверять результат на ошибочное значение. Это легко может увеличить размер программы.

Вариант в) – вернуть допустимое значение и оставить программу в ненормальном состоянии – имеет тот недостаток, что вызывающая функция может не заметить, что программа находится в ненормальном состоянии. Например, многие стандартные функции библиотеки C устанавливают значение глобальной переменной `errno` для индикации ошибки. Однако программы в большинстве случаев не проверяют эту переменную достаточно систематически, чтобы избежать последующих ошибок. Более того, использование глобальных переменных для записи информации об ошибках работает неудовлетворительно при наличии параллельных процессов.

Вариант г) – вызвать функцию, предназначенную для обработки ошибки – возлагает на вызванную функцию решение проблемы одним из перечисленных выше способов.

Механизм обработки исключений предоставляет альтернативу традиционным методам. Он позволяет отделить код обработки ошибок от собственного кода алгоритма, делая программу более понятной и более «чистой». В результате получаем более регулярный способ обработки ошибок, что упрощает взаимодействие между отдельно написанными фрагментами программы.

4.2. Общие принципы обработки исключительных ситуаций в C++

Механизм обработки ситуаций дает способ передачи управления из точки выполнения программы в расположенную выше по управлению точку, в которой определен *обработчик исключений* (exception handler). Главная идея состоит в том, что функция, сталкивающаяся с неразрешимой проблемой, объявит исключительную ситуацию, в надежде на то, что вызвавшая ее (прямо или косвенно) функция может решить проблему. Обработчик ситуации будет вызван только в случае исполнения *выражения-возбуждения-ситуации* внутри так называемого *блока-с-контролем* или в функциях, вызванных из этого блока. В C++ синтаксис возбуждения ситуации выглядит следующим образом:

выражение-возбуждения-ситуации:

throw выражение

Выражение-возбуждения-ситуации в C++ является выражением некоторого типа. *Выражение-возбуждения-ситуации* иногда еще называют точкой возникновения (возбуждения) ситуации (throw-point). О части програм-

мы, в которой исполнилось *выражение-возбуждения-ситуации*, говорят, что в ней возникла ситуация (она возбудила ситуацию).

Рассмотрим, каким образом в C++ можно определить и обработать ошибки диапазона, возникающие в классе Vector.

```
class Vector {
    int* p;
    int* sz;
public:
    class Range { }; // класс ситуаций

    int& operator[] (int i);
    // . . .
};
```

Объекты класса Range предназначены для использования в качестве исключений и возбуждать последние следующим образом:

```
int& Vector::operator[] (int i)
{
    if (0<=i && i<sz) return p[i];
    throw Range();
}
```

В общем случае исключения новых типов следует создавать тогда, когда программист хочет обрабатывать ошибки одного типа и пропускать ошибки других типов.

4.3. Возбуждение ситуации

При возбуждении ситуации управление передается на один из обработчиков ситуации. При этом посылается объект, и тип этого объекта определяет, какой обработчик может перехватить данную ситуацию.

Например, ситуация

```
throw "Help!";
```

может быть перехвачена некоторым обработчиком типа `char*`:

```
try {  
    // . . .  
} catch (const char* p) {  
    // здесь реакция на ситуацию типа символьной строки  
}
```

а ситуация

```
class Overflow {  
    // . . .  
public: Overflow(char, double, double);  
};  
void f(double x) {  
    // . . .  
    throw Overflow('+', x, 3.45e107);  
}
```

может быть перехвачена обработчиком

```
try {  
    // . . .  
    f(1.2);  
    // . . .  
} catch (Overflow& oo) {  
    // здесь обработчик ситуации типа Overflow  
}
```

При возникновении ситуации управление передается на ближайший обработчик соответствующего типа; "ближайший" – тот, в чей *блок-с-контролем* управление попало в последний раз; "соответствующий тип" определяется ниже.

Выражение-возбуждения-ситуации создает временный объект, тип которого статически определяется операндом операции `throw`, и затем инициализирует им переменную, указанную в описании обработчика.

В C++ *выражение-возбуждения-ситуации* без операнда повторно возбуждает обрабатываемую ситуацию. *Выражение-возбуждения-ситуации* без операнда может появиться только в обработчике исключения или в функции, явно или неявно вызванной из него. Например, код, который нужно выполнить при возникновении какой-либо ситуации, не обрабатывая ее полностью, мог бы быть написан так:

```
try {  
    // . . .  
}  
catch (...) { // перехват ситуации  
               // частичная обработка ситуации  
    throw;  
// передача ситуации некоторому другому обработчику  
}
```

4.4. Обработка исключений

Функция, нуждающаяся в обнаружении возбужденной ситуации, должна поместить вызов функции в *блок-с-контролем* с реакцией на ситуацию. Такой блок имеет следующий синтаксис в C++:

блок-с-контролем:

```
try { ... }  
catch (объявлении-ситуации 1) { реакция 1 }  
catch (объявлении-ситуации 2) { реакция 2 }  
catch (...) { реакция n }
```

Например:

```
void f(Vector& v)
{
    // . . .
    try
    {
        do_something(v);
    }
    catch (Vector::Range)
    {
        // реакция на исключение типа Vector::Range
        // так как do_something() вызвало ситуацию,
        // делаем что-то другое,
        // попадаем сюда только, если обращение к
        // do_something() приводит
        // к вызову Vector::operator[](i) с плохим
        // индексом
    }
    // . . .
}
```

Конструкция

```
catch (/* . . . */) { }
```

представляет собой обработчик ситуации. Он может использоваться только сразу за *блоком-с-контролем* или непосредственно за другим обработчиком (если есть несколько обработчиков разных типов).

Многоточие (. . .) в *объявлении-ситуации* дает отождествление для любой ситуации. Обработчик с многоточием, если он есть, должен быть последним в своем *блоке-с-контролем*.

5. РЕАЛИЗАЦИЯ ПРИМЕРОВ

5.1. Класс полином

```
class stp          // Базовый класс
{ protected:
  int n;
  stp(){};
  stp(int st)
    {n=st;}
};
```

Класс полином poli является производным классом.

```
class poli: public stp
{ public:
  int *a;
  poli(){};
  poli(int s){ n=s;};
  ~poli()    { delete []a;};
  void cr_poli();
  poli(poli &y);
  int operator()(int );
  friend poli operator+ (poli bb, poli cc);
  poli & operator= (poli bb);
  void prosm();
};

poli & poli:: operator= (poli bb)
{ n=bb.n;
  for (int i=0;i<=n;i=i+1) a[i]=bb.a[i];
  return *this;
};

poli::poli(poli &y)
{ n=y.n;  a=new int[n+1];
  for(int i=0;i<=n;i++)
    a[i]=y.a[i];
};
```

```

void poli::cr_poli()
{ a=new int[n+1];
  cout<<"ВВЕДИТЕ КОЭФ-ТЫ "<<n;
  for (int i=0;i<=n;i++)
    cin>>a[i];
}

int poli:: operator()(int x)
{ int k=n;
  float r;
  r=a[0];
  for (int i=1;i<=k;i=i+1)
    r=r*x+a[i];
  return r;
}

poli operator+ (poli bb,poli cc)
{ int i,k,l;
  k=bb.n; l=cc.n;
  if (k>=l)
  { poli r(k);
    for (i=0;i<=k;i++) r.a[i]=bb.a[i];
    for (i=0;i<=l;i++)
      r.a[k-l+i]=r.a[k-l+i]+cc.a[i];
    return r;
  };
  poli t(l);
  for (i=0;i<=l;i++) t.a[i]=cc.a[i];
  for (i=0;i<=k;i++)
    t.a[l-k+i]=t.a[l-k+i]+bb.a[i];
  return t;
}

void poli:: prosm()
{ int i;
  for (i=0;i<=n;i++)
    cout <<a[i]<<" ";
  cout<<"\n";
} ;

```

5.2. Класс таблица

```
class zap      // Базовый класс
{ protected:
  char fio[31];
  char pr[10];
  int a1;
public:
  zap(){};
  zap(zap &z)
  { strcpy(fio,z.fio);
    strcpy(pr,z.pr);    a1=z.a1;
  };
  int get(){ return a1;};
  char* get_f()
  { char *s; s=strdup(fio);
    return s;
  };
  void input();
  void output();
};

class table : public zap
{ public:
  zap *a; int n;
  table(){};
  table(int s){ n=s;};
  ~table()    { delete []a;};
  void cr_table();
  void sort(int t);
  void sort(char t);
  void show();
};

void zap::input()
{ cout<<"\nVvod fio\n";    cin>>fio;
  cout<<"\nVvod pred\n";   cin>>pr;
  cout<<"\nVvod ball\n";   cin>>a1;
};

void zap::output()
{ cout<< fio <<" " <<pr<<" " <<a1<<" " ;
```

```

};
void table :: cr_table()
{ a=new zap [n];
  for(int i=0;i<n;i++)
    a[i].input();
  cout <<"\n таблица создана \n";
};
void table :: show()
{ int i;
  for(i=0;i<n;i++)
  { a[i].output();
    cout <<"\n ";
  };
};
int SRAVN(int a,int b)
{ if(a>b) return 1;
  else return 0;
};
int SRAVN(char a[],char b[])
{ char *u,*v;
  u=strdup(a);v=strdup(b);
  if (strcmp(u,v)>0) return 1;
  else return 0;
};
void table :: sort(char t)
{ zap s; int i,j;
  for (i=n-1;i>0;i--)
  { int k=0;
    for (j=1;j<i;j++)
      if (SRAVN(a[j].get_f(),a[k].get_f())) k=j;
    s=a[k];a[k]=a[i];a[i]=s;
  }
};
void table :: sort(int t)
{ zap s;
  int i,j;
  for (i=n-1;i>0;i--)
  { int k=0;
    for (j=1;j<i;j++)
      if (SRAVN(a[j].get(),a[k].get())) k=j;
    s=a[k];a[k]=a[i];a[i]=s;
  }
};

```

}

6. ЗАДАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

1) Описать класс «Дробное число со знаком». Число должно быть представлено двумя полями: целая часть – длинное целое число со знаком, дробная часть – беззнаковое короткое целое число. Реализовать арифметические операции сложения, вычитания, умножения, деления и операцию сравнения дробных чисел. Написать текст программы на языке C++, демонстрирующей функционал объектов, порожденных от класса.

2) Описать класс «Деньги» для работы с денежными суммами. Денежная сумма должна быть представлена двумя полями: длинное целое число для рублей и беззнаковое короткое целое число – для копеек. Дробная часть денежной суммы (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение сумм, вычитание сумм, деление сумм, деление суммы на дробное число, умножение суммы на дробное число, операцию сравнения сумм и преобразование денежной суммы в символьную строку. Написать текст программы на языке C++, демонстрирующей функционал объектов, порожденных от класса.

3) Описать класс «Трапеция», поля класса – координаты четырех точек на плоскости: координаты – числа с плавающей запятой. Предусмотреть в классе конструктор и методы: проверка, является ли фигура, образованная четырьмя точками трапецией; проверка, является ли фигура, образованная четырьмя точками равнобедренной трапецией, вычисление длины каждой стороны, вычисление периметра трапеции, вычисление площади трапеции. Написать текст программы на языке C++, демонстрирующей функционал объекта, порожденного от класса.

4) Описать класс «Комплексное число», поля класса - действительная и мнимая части комплексного числа являются числами с плавающей запятой и находятся в закрытом разделе класса. Предусмотреть в классе конструктор и методы: присваивание комплексному числу значения, сложение комплексных чисел, вычитание комплексных чисел, умножение комплексных чисел, деление комплексных чисел и преобразование комплексного числа в символьную строку. Написать текст программы на языке C++, демонстрирующей функционал объектов, порожденных от класса.

ЛИТЕРАТУРА

1. Невская, Е.С. Искусство программирования [Текст]: Учеб. пособие для вузов / Е.С. Невская, А.А. Чекулаева, М.И. Чердынцева – Москва: Вузовская книга, 2002. - 207 с.

2. Страуструп, Б. Язык программирования С++ для профессионалов [Текст] / Б. Страуструп. – Москва: Интернет-Университет Информационных Технологий, 2006. – 568 с.

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА
Б1.В.09.11 КОНСТРУИРОВАНИЕ МЕХАТРОННЫХ
МОДУЛЕЙ**

Специальность –
21.05.04 Горное дело
Специализация –

Мехатроника и робототехника промышленных комплексов

Авторы: Таугер В.М.

Одобрена на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой

(подпись)

Волков Е.Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического

(название факультета)

Председатель

(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

Задания на курсовой проект составлены в соответствии с учебным планом по дисциплине «Конструирование мехатронных модулей» для студентов специальности 21.05.04 Горное дело, специализация «Мехатроника и робототехника промышленных комплексов».

Задания представляют собой 32 варианта технических требований к мехатронному модулю. Каждый проектант должен сконструировать мехатронный модуль в соответствии с определенным вариантом технических требований.

Составитель: В.М. Таугер, канд. техн. наук, доцент кафедры «Техническая механика».

ОГЛАВЛЕНИЕ

1.ОСНОВНЫЕ ПОЛОЖЕНИЯ	4
2.ОБЩИЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ	5
3.СПЕЦИАЛЬНЫЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ	6
4.ВАРИАНТЫ ТЕХНИЧЕСКИХ ЗАДАНИЙ	9
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	10

1. ОБЩИЕ ТРЕБОВАНИЯ

Задачей курсового проектирования является конструирование мехатронного модуля в соответствии с техническим заданием.

Техническое задание включает в себя общие и специальные технические требования. Общие требования (раздел 2) относятся ко всем подлежащим разработке мехатронным модулям. Конкретные сочетания специальных требований (раздел 3) различны для каждого варианта. Ведущий преподаватель назначает вариант из табл. 3.1 проектанту персонально.

Руководствуясь методикой проектирования мехатронных модулей, студент формирует последовательно *F*-, *S*- и *C*-модели устройства, а затем для окончательно принятой *C*-модели разрабатывает техническую документацию в составе:

- расчетно-пояснительная записка;
- чертеж общего вида мехатронного модуля.

Расчетно-пояснительная записка должна содержать: обоснование основных конструктивных решений; сравнительную оценку стоимости изготовления различных вариантов устройства модуля; расчет преобразователя движения, включающий в себя определение передаточного отношения и проверки на прочность и кинематическую точность; схемы, алгоритмы и т.п., необходимые для пояснения устройства и работы мехатронного модуля.

Расчетно-пояснительную записку следует выполнять на листах формата А4 с соблюдением правил ЕСКД к текстовым документам.

Чертеж общего вида должен содержать виды и разрезы с размерами, дающие полное представление о конструкции и технических возможностях мехатронного модуля и обеспечивающие возможность дальнейшей разработки чертежно-технической документации, а также техническую характеристику модуля. На свободном поле чертежа должна быть дана экспликация с перечнем основных узлов и важнейших деталей изделия.

Чертеж общего вида следует выполнять на листе формата А1 (или большем - при необходимости) с соблюдением соответствующих требований ЕСКД.

Консультации в процессе выполнения и прием курсового проекта осуществляет ведущий преподаватель.

2. ОБЩИЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

2.1. Низкая трудоемкость монтажа (демонтажа), сборки (разборки), технического контроля и обслуживания.

2.2. Функционирование в автоматическом режиме. Конструкция должна обеспечивать изменение скорости и направления движения рабочего органа.

2.3. Автоматизированный контроль технических параметров, самодиагностика неисправностей.

2.4. Компактность конструкции.

2.5. Минимальные массогабаритные показатели.

2.6. Предпочтительно применение асинхронных электродвигателей с короткозамкнутым ротором.

2.7. Движение рабочего органа под действием нагрузки при выключенном двигателе не допускается. В мехатронных модулях с самотормозящимися преобразователями движения тормозной момент на валу двигателя должен составлять не менее 20% от крутящего момента при номинальной нагрузке.

3. СПЕЦИАЛЬНЫЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

3.1. Вид движения рабочего органа:

- а) вращательное, ось вращения вертикальная;
- б) вращательное, ось вращения горизонтальная;
- в) возвратно-поступательное горизонтальное;
- г) возвратно-поступательное вертикальное.

3.2. Максимальная величина перемещения рабочего органа:

- а) неограниченное число оборотов;
- б) два оборота;
- в) 1,5л;
- г) 1000 мм;
- д) 500 мм;
- е) 100 мм.

3.3. Характер нагрузки на рабочем органе:

- а) реверсивная;
- б) нереверсивная.

3.4. Номинальная величина нагрузки на рабочем органе:

- а) $T_{\text{ном}} = 1000 \text{ Нм}$;
- б) $T_{\text{ном}} = 200 \text{ Нм}$;
- в) $T_{\text{ном}} = 50 \text{ Нм}$;
- г) $F_{\text{ном}} = 10 \text{ кН}$;
- д) $F_{\text{ном}} = 2 \text{ кН}$;
- е) $F_{\text{ном}} = 0,5 \text{ кН}$.

Обозначения: $T_{\text{ном}}$ – номинальный крутящий момент; $F_{\text{ном}}$ – номинальная осевая сила.

3.5. Номинальная скорость движения рабочего органа при номинальной нагрузке:

- а) $\omega_{\text{ном}} = 7,85 \text{ с}^{-1}$;
- б) $\omega_{\text{ном}} = 3,93 \text{ с}^{-1}$;
- в) $\omega_{\text{ном}} = 0,785 \text{ с}^{-1}$;
- г) $v_{\text{ном}} = 1 \text{ м/с}$;
- д) $v_{\text{ном}} = 0,2 \text{ м/с}$;
- е) $v_{\text{ном}} = 0,05 \text{ м/с}$.

Обозначения: $\omega_{\text{ном}}$ – угловая скорость; $v_{\text{ном}}$ – линейная скорость.

3.6. Способ крепления модуля:

- а) фланцевый;
- б) на лапах.

3.7. Режим эксплуатации:

- а) по рис. 3.7.1;
- б) по рис. 3.7.2;
- в) по рис. 3.7.3.

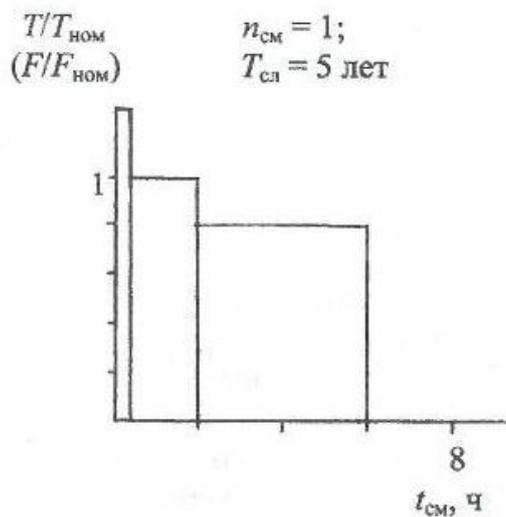


Рис. 3.7.1

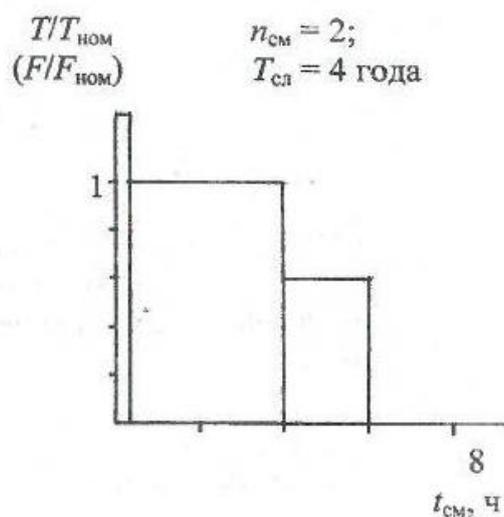


Рис. 3.7.2

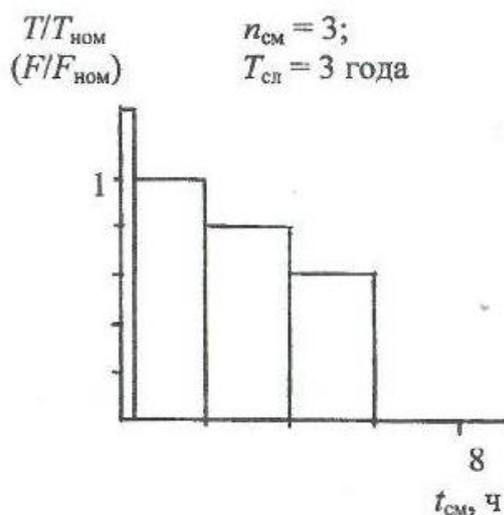


Рис. 3.7.3

Обозначения: T, F – значения нагрузки; $n_{\text{см}}$ – число смен в сутки; $t_{\text{см}}$ – продолжительность смены; $T_{\text{сл}}$ – срок службы модуля.

3.8. Допустимая кинематическая погрешность преобразователя движения:

- а) $[F'_{\text{iop}}] = 0,15 \text{ мм};$
- б) $[F'_{\text{iop}}] = 0,08 \text{ мм};$
- в) $[F'_{\text{iop}}] = 0,02 \text{ мм};$

- г) $[\delta\varphi] = 0,0015$ рад;
- д) $[\delta\varphi] = 8 \cdot 10^{-4}$ рад;
- е) $[\delta\varphi] = 2 \cdot 10^{-4}$ рад.

3.9. Допустимая величина мертвого хода:

- а) $[J_{тp}] = 0,3$ мм;
- б) $[J_{тp}] = 0,1$ мм;
- в) $[J_{\varphi p}] = 0,003$ рад;
- г) $[J_{\varphi p}] = 0,001$ рад.

3.10. Дополнительные требования:

- а) самоторможение рабочего органа;
- б) минимальная высота (вертикальный габарит) мехатронного модуля;
- в) минимальная длина (наибольший горизонтальный габарит) мехатронного модуля;
- г) функционирование при температурах воздуха $-30 \dots +20$ °С;
- д) соосность вала двигателя и выходного (тихоходного) вала модуля.

4. ВАРИАНТЫ ТЕХНИЧЕСКИХ ЗАДАНИЙ

4.1. Общие технические требования к мехатронному модулю см. раздел 2.

4.2. Специальные технические требования см. табл. 4.1.

Таблица 4.1

Варианты комплексов специальных технических требований
к мехатронному модулю

Номер варианта	Специальные технические требования									
	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	3.10
1	а	а	а	а	а	а	а	а	б	б
2	б	б	б	б	б	б	б	б	-	а
3	в	д	а	г	г	а	в	г	в	-
4	г	е	б	д	д	б	а	д	-	г
5	а	в	а	в	в	а	б	в	а	д
6	б	а	б	а	а	б	в	а	-	в
7	в	е	а	е	е	а	а	е	г	а
8	г	д	б	г	г	б	б	г	-	-
9	а	б	а	б	б	а	в	б	б	а
10	б	в	б	в	в	б	а	в	-	г
11	в	г	а	д	д	а	б	д	г	-
12	г	г	б	е	е	б	в	е	-	а
13	а	а	а	в	в	а	а	в	а	д
14	б	б	б	а	а	б	б	а	-	-
15	в	г	а	г	г	а	в	г	в	в
16	г	д	б	е	е	б	а	е	-	а
17	а	а	б	а	а	б	б	а	-	г
18	б	а	а	б	б	а	в	б	б	-
19	в	д	б	д	д	б	а	д	-	а
20	г	г	а	г	г	а	б	г	в	б
21	а	б	б	а	в	б	в	б	-	д
22	б	в	а	в	а	а	а	в	а	в
23	в	е	б	е	д	б	б	е	-	-
24	г	д	а	д	д	а	в	д	г	-
25	а	в	б	б	б	б	а	б	-	-
26	б	б	а	а	а	а	б	а	б	в
27	в	г	б	г	е	б	в	г	-	-
28	г	е	а	е	е	а	а	г	в	а
29	а	в	б	а	а	б	б	в	-	а
30	б	а	а	б	в	а	в	в	а	д
31	в	д	б	е	г	б	а	д	-	в
32	г	г	а	д	г	а	б	е	г	-

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Егоров О.Д., Подураев Ю.В. Мехатронные модули. Расчет и конструирование: Учебное пособие. -М.: МГТУ «СТАНКИН», 2004.
2. Таугер В.М. Основы конструирования мехатронных модулей и систем: Учебное пособие. - Екатеринбург: УрГУПС, 2004.
3. Таугер В.М. Конструирование преобразователей движения мехатронных модулей: Учебное пособие. - Екатеринбург: УрГУПС, 2006.

УТВЕРЖДАЮ
Проректор по учебно-методической
работе _____ В. В. Зубов



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Б1.В.09.12 SCADA-СИСТЕМЫ

Специальность -
21.05.04 Горное дело

Специализация -
Мехатроника и робототехника промышленных комплексов

Одобрена на заседании кафедры

Автоматики и компьютерных
технологий

(название кафедры)

Зав.
кафедрой

(подпись)

Бочков В. С.

(Фамилия И.О.)

Протокол № 1 от 19.09.2024

(Дата)

Рассмотрена методической
комиссией
факультета

горно-механического

(название факультета)

Председатель

(подпись)

Осипов П. А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

ЕКАТЕРИНБУРГ

Содержание

Содержание.....	2
Часть 1. SCADA-система Advantech GeniDAQ	8
Глава 1. Обзор SCADA-систем	9
1.1. Введение.....	9
1.2. Функциональная структура SCADA-системы	11
1.3. Особенности SCADA как процесса управления.....	11
1.4. Основные требования к SCADA-системам.....	12
1.5. Функциональные возможности SCADA-систем	13
1.6. Программные платформы SCADA-систем	13
1.7. Средства сетевой поддержки SCADA-систем	15
1.8. Встроенные языки программирования SCADA-систем	15
1.9. Базы данных.....	17
1.10. Организация взаимодействия с устройствами нижнего уровня	18
1.11. Открытость SCADA-систем	19
1.12. О реальном времени.....	20
1.13. Средства визуализации	22
1.14. Состояние тревоги и события	23
1.15. Отображение и архивирование данных.....	23
1.16. Стоимость системы	24
1.17. SCADA-продукты на российском рынке	25
1.17.1. Базовые свойства	25
1.17.2. Возможности среды разработки	26
1.17.3. Возможности конфигурирования системы	27
1.17.4. Характеристики сопровождения/эксплуатации.....	27
1.17. Заключение	27
Глава 2. OPC — промышленный стандарт и средство интеграции компонентов в промышленной автоматизации	30
2.1. Что такое OPC?.....	31
2.2. DCOM и OPC-приложения.....	32
2.3. Почему OPC?	35
2.4. Заключение	36
Глава 3. Краткий обзор SCADA-системы GeniDAQ.....	37
3.1. Почему мы рассматриваем GeniDAQ?	37
3.2. Системная архитектура GeniDAQ	37
Глава 4. Учебник по SCADA-системе GeniDAQ.....	41
4.1. Занятие 1 "Демонстрация базовых приемов работы — одна задача реального времени с отображением результатов ее работы в одном экранном окне"	42
4.1.1. Используемый инструментарий.....	42
4.1.2. Проектирование приложения.....	47
4.1.3. Упражнения	53

4.2. Занятие 2 "Переключение окон отображения"	55
4.2.1. Используемый инструментарий.....	55
4.2.2. Проектирование приложения.....	56
4.2.3. Упражнения	60
4.3. Занятие 3 "Просмотр и изменение порядка выполнения функциональных блоков редактора задач"	64
4.3.1. Используемый инструментарий.....	64
4.3.2. Проектирование приложения.....	67
4.3.3. Упражнения	71
4.4. Занятие 4 "Использование инструментов рисования редактора форм отображения"	72
4.4.1. Используемый инструментарий.....	73
4.4.2. Проектирование приложения.....	81
4.4.3. Упражнения	83
4.5. Занятие 5 "Использование тега для связи между задачей и формой отображения"	85
4.5.1. Используемый инструментарий.....	85
4.5.2. Проектирование приложения.....	92
4.5.3. Упражнения	96
4.6. Использование языка VBA в SCADA-системах.....	98
4.6.1. Средства языка VBA, инвариантные к среде использования (Microsoft Visual Basic 6.3)	98
4.6.2. Использование языка VBA в SCADA-системе GeniDAQ	143
4.6.3. Упражнения	164
4.7. Занятие 6 "Использование функционального блока Бейсик-сценария"	167
4.7.1. Используемый инструментарий.....	167
4.7.2. Проектирование приложения.....	168
4.7.3. Упражнения	169
4.8. Занятие 7 "Совместное использование функциональных блоков Бейсик-сценария и виртуального тега"	175
4.8.1. Используемый инструментарий.....	175
4.8.2. Проектирование приложения.....	178
4.8.3. Упражнения	181
4.9. Занятие 8 "Программирование основного сценария"	183
4.9.1. Используемый инструментарий.....	184
4.9.2. Проектирование приложения.....	184
4.9.3. Упражнения	185
4.10. Занятие 9 "Управление несколькими задачами"	186
4.10.1. Используемый инструментарий.....	187
4.10.2. Проектирование приложения.....	187
4.10.3. Упражнения	190
4.11. Занятие 10 "DDE-обмен с использованием блока Бейсик-сценария редактора задач"	194
4.11.1. Проектирование приложения.....	195
4.11.2. Упражнения	201
4.12. Занятие 11 "Работа с таймером".....	202

4.12.1. Используемый инструментарий.....	202
4.12.2. Проектирование приложения.....	205
4.11.2. Упражнения	206
Глава 5. OPC-технология информационной интеграции SCADA-системы и промышленного контроллера	211
5.1. Промышленный контроллер 7188E2 фирмы ICP DAS (Тайвань).....	211
5.2. Интерфейс OPC-сервера OpenLabOPCSvr.AiVT.2 и его конфигурирование ..	216
5.3. Настройка компьютеров локальной вычислительной сети	219
5.3.1. Настройка ОС Windows2000 для работы с OPC-серверами (на примере OPC-сервера OpenLabOPCSvr.AiVT.2)	219
5.3.2. Настройка ОС Windows XP для работы с OPC-серверами (на примере OPC-сервера OpenLabOPCSvr.AiVT.2)	220
5.4. Занятие 12 "Информационный обмен между SCADA-системой и промышленным контроллером".....	223
5.4.1. Используемый инструментарий.....	224
5.4.2. Проектирование приложения.....	229
5.4.3. Упражнения	235
Литература	247



Часть 1. SCADA-система Advantech GeniDAQ

Глава 1. Обзор SCADA-систем

**Глава 2. OPC — промышленный стандарт и
средство интеграции компонентов в
промышленной автоматизации**

**Глава 3. Краткий обзор SCADA-системы
GeniDAQ**

Глава 4. Учебник по SCADA-системе GeniDAQ

**Глава 5. OPC-технология информационной
интеграции SCADA-системы и промышленного
контроллера**

Глава 1. Обзор SCADA-систем

За последние 10-15 лет за рубежом и в нашей стране резко возрос интерес к проблемам построения высоконадежных и высокоэффективных систем диспетчерского управления и сбора данных, получивших название *SCADA-систем* (SCADA — Supervisory Control And Data Acquisition, диспетчерское управление и сбор данных). Что же понимают под SCADA-системой и какими свойствами и особенностями она обладает?

1.1. Введение

Современное производство не может обойтись без *автоматизации*, которая позволяет повысить качество, снизить процент брака, увеличить выход продукции, снизить себестоимость продукции, сэкономить ресурсы и продлить срок службы технологического оборудования. В настоящее время автоматизированные системы управления промышленными предприятиями и технологическими процессами (*АСУ ТП*) имеют иерархическую структуру, включающую следующие неотъемлемые уровни [1]:

- нижний уровень* — технологические устройства (датчики, контроллеры, исполнительные механизмы и т.д.);
- средний уровень* — SCADA-системы;
- верхний уровень* — приложения, управляющие ресурсами предприятия.

Совет

Внимательно изучите все описанные далее характеристики и особенности SCADA-систем — это важно.

В рамках данного учебного курса мы будем интересоваться SCADA-системами. Диспетчерское управление и сбор данных является основным и перспективным методом автоматизированного управления сложными динамическими системами в жизненно важных и критичных областях. На основе SCADA-систем строятся крупные автоматизированные системы управления в промышленности, энергетике, на транспорте, в космической, военной областях и в различных государственных структурах. SCADA-система реализует процесс сбора информации в реальном времени с удалённых объектов для обработки, анализа и управления такими объектами. Пробразом современных SCADA-систем на ранних стадиях развития автоматизированных систем управления являлись системы телеметрии и сигнализации. SCADA-система — это совокупность устройств управления и мониторинга, а также средств их взаимодействия с технологическим объектом. Под термином SCADA понимают набор программных и аппаратных средств для реализации операторских рабочих мест, а поскольку программные средства являются здесь определяющими, то зачастую о SCADA-системах говорят лишь как о программном обеспечении.

Как отмечалось ранее, диспетчерское управление и сбор данных является основным и в настоящее время остается наиболее перспективным методом автоматизированного управления сложными динамическими системами (процессами) в жизненно важных и критичных с точки зрения безопасности и надежности областях [2]. За последние 10-15 лет за рубежом и в нашей стране резко возрос интерес к проблемам построения высоконадежных и высокоэффективных систем диспетчерского управления и сбора данных. С одной стороны, это связано со значительным прогрессом в области вычислительной техники, программного обеспечения и телекоммуникаций, что увеличивает возможности и расширяет сферу применения автоматизированных систем. С другой стороны, развитие информационных технологий, повышение степени автоматизации и перераспределение функций между человеком и аппаратурой обострило проблему взаимодействия человека-оператора с системой управления. Расследование и анализ большинства аварий и происшествий в авиации, наземном и водном транспорте, промышленности и энергетике, часть из которых привела к катастрофическим последствиям, показало следующее. Если в 60-х годах ошибка человека являлась первоначальной причиной лишь 20% инцидентов (80%, соответственно, за технологическими неисправностями и отказами), то в 90-х годах доля "человеческого фактора" возросла до 80%, причем, в связи с постоянным совершенствованием технологий и повышением надежности электронного оборудования и машин, доля эта может еще возрасти.

Основной причиной такой тенденции является старый, традиционный подход к построению сложных автоматизированных систем управления, который часто применяется и в настоящее время. Это ориентация, в первую очередь, на применение новейших технических (технологических) достижений, стремление повысить степень автоматизации и функциональные возможности системы и, в то же время, недооценка необходимости построения эффективного *человеко-машинного интерфейса*, ориентированного на пользователя (оператора). Не случайно, что именно на последние 15 лет, т. е. на период появления мощных, компактных и недорогих вычислительных средств, пришелся пик исследований в США по проблемам человеческого фактора в системах управления, в том числе по оптимизации архитектуры и НМИ-интерфейса систем диспетчерского управления и сбора данных.

Изучение материалов по проблемам построения эффективных и надежных систем диспетчерского управления показало необходимость применения нового подхода при разработке таких систем (*human-centered design*), ориентированного в первую очередь на человека-оператора (диспетчера), вместо традиционного и повсеместно применявшегося подхода "*hardware-centered*", при котором при построении системы основное внимание уделялось выбору и разработке технических средств и программного обеспечения. Применение нового подхода привело к созданию SCADA-систем. Использование их в реальных космических и авиационных разработках и сравнительные испытания систем в Национальном управлении по авионавтике и исследованию космического пространства США подтвердили их эффективность, позволив увеличить производительность операторов, на порядок

уменьшить процедурные ошибки и свести к нулю критические (некорректируемые) ошибки операторов.

1.2. Функциональная структура SCADA-системы

Существует два типа управления удаленными объектами в SCADA-системах: автоматическое и инициируемое оператором системы [2]. В SCADA-системах можно выделить четыре основных функциональных компонента — человек-оператор, компьютер взаимодействия с человеком (рабочая станция), компьютер взаимодействия с задачей (объектом) и задача (объект управления). Человеку-оператору в SCADA-системе присущи пять основных функций, которые можно рассматривать как набор вложенных циклов, в которых оператор:

- планирует, какие следующие действия необходимо выполнить;
- обучает (программирует) компьютерную систему на последующие действия;
- отслеживает результаты полуавтоматической работы системы;
- вмешивается в процесс в случае критических событий, когда автоматика не может справиться, либо, при необходимости, подстраивает параметры процесса;
- обучается в процессе работы (получает опыт).

Данное представление SCADA-системы явилось основой для разработки современных методологий построения эффективных диспетчерских систем.

1.3. Особенности SCADA как процесса управления

Процессу управления в современных диспетчерских системах присущи следующие особенности [2]:

- процесс SCADA применяется в системах, в которых обязательно наличие человека (оператора, диспетчера);
- процесс SCADA был разработан для систем, в которых любое неправильное воздействие может привести к отказу (потере) объекта управления или даже катастрофическим последствиям;
- оператор несет, как правило, общую ответственность за управление системой, которая, при нормальных условиях, только изредка требует подстройки параметров для достижения оптимальности;
- активное участие оператора в процессе управления происходит нечасто и в непредсказуемые моменты времени, обычно в случае наступления критических событий (отказы, нештатные ситуации и т.п.);
- действия оператора в критических ситуациях могут быть жестко ограничены по времени (несколькими минутами или даже секундами).

1.4. Основные требования к SCADA-системам

К SCADA-системам предъявляются следующие основные требования.

- Надёжность системы (технологическая и функциональная). Никакой единичный отказ или единичная ошибка не должны вызывать выдачу ложного выходного воздействия.
- Безопасность управления.
- Точность обработки и представления данных.
- Простота расширения системы.
- Интуитивность, простота использования.

Требования безопасности и надёжности управления в SCADA-системах сводятся к следующему. Никакой единичный отказ оборудования не должен вызывать выдачу ложного выходного воздействия (команды) на объект управления. Никакая единичная ошибка оператора не должна вызывать выдачу ложного выходного воздействия (команды) на объект управления. Все операции по управлению должны быть интуитивно-понятными и удобными для оператора (диспетчера).

В силу требований, предъявляемых к SCADA-системам, спектр их функциональных возможностей определен и реализован практически во всех SCADA-пакетах. Перечислим основные возможности и средства, присущие всем SCADA-системам и различающиеся только техническими особенностями реализации:

- автоматизированная разработка, дающая возможность создания программного обеспечения (ПО) системы автоматизации без реального программирования, т.е. без написания программного кода в классическом смысле;
- непосредственное автоматическое управление технологическим процессом в соответствии с заданными алгоритмами;
- средства приёма первичной информации о контролируемых технологических параметрах от контроллеров нижних уровней и датчиков;
- средства управления и регистрации сигналов в аварийных ситуациях;
- средства приёма команд оператора и передача их исполнительным механизмам и контроллерам нижних уровней;
- средства регистрации событий, связанных с контролируемым технологическим процессом и действиями персонала, ответственного за эксплуатацию и обслуживание системы;
- средства хранения информации с возможностью ее постобработки;
- средства вторичной обработки принятой информации;
- средства визуализации информации о технологическом процессе и архивной информации в удобной для восприятия форме;
- средства формирования сводок и других отчётных документов на основе архивной информации;

- средства обмена информацией с автоматизированной системой управления предприятием;
- возможность работы прикладной системы с наборами параметров, рассматриваемыми как "единое целое".

Если попытаться коротко охарактеризовать основные функции, то можно сказать, что SCADA-система собирает информацию о технологическом процессе, обеспечивает интерфейс с оператором, сохраняет историю процесса и осуществляет автоматическое управление процессом в том объеме, в котором это необходимо и присуще всем SCADA-системам.

1.5. Функциональные возможности SCADA-систем

В основе большинства SCADA-пакетов лежит использование нескольких программных компонентов (базы данных реального времени, ввода-вывода, предыстории, аварийных ситуаций) и программных администраторов (доступа, управления, сообщений). Следует отметить, что в целом технологии проектирования систем автоматизации на основе SCADA-систем очень похожи:

- Разработка архитектуры системы автоматизации в целом. На этом этапе определяется функциональное назначение каждого узла системы автоматизации.
- Решение вопросов, связанных с возможной поддержкой распределенной архитектуры, необходимостью введения узлов с "горячим резервированием" и т. п.
- Создание прикладной системы управления для каждого узла. На этом этапе специалист в области автоматизируемых процессов наполняет узлы архитектуры алгоритмами, совокупность которых позволяет решать задачи автоматизации.
- Приведение в соответствие параметров прикладной системы с информацией, которой обмениваются устройства нижнего уровня (например, программируемые логические контроллеры) с внешним миром (датчики температуры, давления и др.).
- Отладка созданной прикладной программы в режиме эмуляции и в реальном режиме.

Перечисленные выше возможности SCADA-систем в значительной мере определяют стоимость и сроки создания ПО, а также сроки ее окупаемости.

1.6. Программные платформы SCADA-систем

Программно-аппаратные платформы, на которых реализованы SCADA-системы, являются важной характеристикой для оценки их функциональности. Анализ таких платформ необходим, поскольку от него зависит ответ на вопросы распространения SCADA-системы на имеющиеся вычислительные средства, а также оценка стоимости эксплуатации системы. В различных SCADA-системах этот вопрос решен по разному

[3, 4]. Так, SCADA-система FactoryLink (United States DATA Co.) имеет весьма широкий список поддерживаемых программно-аппаратных платформ: DOS, OS/2, Unix, VMS, AIX, Windows NT/2K/XP. В то же самое время, в таких SCADA-системах, как RealFlex (BJ Software Systems), Sitex (Jade Software) основу программной платформы принципиально составляет единственная операционная система реального времени QNX. Подавляющее большинство SCADA-систем реализовано на MS Windows платформах. Именно такие системы предлагают наиболее полные и легко наращиваемые MMI-средства (Man Machine Interface). Учитывая продолжающееся усиление позиций фирмы Microsoft на рынке операционных систем (ОС) следует отметить, что даже разработчики многоплатформенных SCADA-систем, такие как United States DATA Co, приоритетным считают дальнейшее развитие своих SCADA-систем на платформе Windows (табл. 1.1).

Таблица 1.1. SCADA-системы на российском и западном рынках

SCADA-системы	Фирма-изготовитель	Страна	2K/XP поддержка	OPC
InTouch	Wonderware	США	✓	✓
GeniDAQ	Advantech	США	✓	✓
Advantech FX	Advantech	США	✓	✓
Genesis32	Iconics	США	✓	✓
Trace Mode	AdAstra	Россия	✓	✓
Citect	Citect Pty. Ltd	Австралия	✓	✓
Factory Link	US DATA	США	✓	✓
LabView	National Instruments	США	✓	✓
RSView	Rockwell Software	США	✓	✓
RealFlex	BJ Software Systems	США	-	-
Sitex	Jade Software	Англия	-	-
iFIX	Intelution	США	✓	✓
MasterSCADA	inSAT	Россия	✓	✓
PI System	OSI Software	США	✓	✓
Контур	Объединение Юг	Украина	✓	✓

Некоторые фирмы, до сих пор поддерживавшие SCADA-системы на базе ОС реального времени (PB), начали менять ориентацию, выбирая системы на платформе Windows. Фирма BJ Software Systems объявила о том, что SCADA-пакет RealFlex дальше развиваться не будет и перевод его на платформу Photon не планируется. Все более очевидным становится применение ОС реального времени, в основном, во встраиваемых системах, где они действительно хороши. Таким образом, основным полем, где сегодня разворачиваются главные события глобального рынка SCADA-систем, стали ОС MS Windows NT/2K/XP. Быстрое развитие OPC-технологий (см. далее подразд. 1.10), низкие цены аппаратного обеспечения,

распространённость ОС Windows на офисных рынках с её солидными техническими характеристиками — главные причины того, что абсолютное большинство производителей SCADA-пакетов мигрировали в сторону этой операционной системы.

1.7. Средства сетевой поддержки SCADA-систем

Одной из основных черт современного мира систем автоматизации является их высокая степень интеграции. В любой из них могут быть задействованы объекты управления; исполнительные механизмы; аппаратура, регистрирующая и обрабатывающая информацию; рабочие места операторов; серверы баз данных и т. д.

Очевидно, что для эффективного функционирования в этой разнородной аппаратной среде SCADA-система должна обеспечивать высокий уровень сетевого сервиса. Желательно, чтобы она поддерживала работу в стандартных сетевых средах (Arcnet, Ethernet и т. д.) с использованием стандартных протоколов (Netbios, TCP/IP и др.), а также обеспечивала поддержку наиболее популярных сетевых стандартов из класса промышленных интерфейсов (Profibus, CANbus, Lon, Modbus и т. д.). Этим требованиям в той или иной степени удовлетворяют практически все рассматриваемые SCADA-системы, с тем только различием, что набор поддерживаемых сетевых интерфейсов, конечно же, в различных SCADA-системах разный. Сетевые средства SCADA-системы GeniDAQ используют протокол TCP/IP, который обеспечивает возможность работы с данными технологического процесса в реальном времени с любого узла сети, а также дистанционное управление процессом. Сетевые средства SCADA-систем Citect, Trace Mode поддерживаются такими сетевыми протоколами, как NetBEUI, IPX/SPX, TCP/IP.

1.8. Встроенные языки программирования SCADA-систем

Встроенные языки программирования — мощное средство SCADA-систем, облегчающее процесс реализации сложных алгоритмов обработки и анализа данных. Кроме того, встроенные языки программирования являются мощным и универсальным средством адаптации систем к требованиям прикладной задачи.

Первые версии SCADA-систем либо не имели подобных языков, либо эти языки реализовывали небогатый набор функций. В современных версиях SCADA-систем функциональные возможности языков становятся существенно богаче. Явно выделяются два подхода.

- Ориентация встроенных языков программирования на технологов. Функции в таких языках являются высокоуровневыми, не требующими профессиональных навыков программирования при их использовании. Количество таких функций в базовых поставках относительно невелико, хотя существуют свободно распространяемые библиотеки дополнительных функций.

- Ориентация на системных интеграторов. В этом случае в качестве языков чаще всего используются Visual Basic-подобные языки.

В каждом языке допускается расширение набора функций. В языках, ориентированных на технологов, это расширение достигается с помощью дополнительных инструментальных средств (Toolkits). Разработка дополнительных функций выполняется обычно программистами-профессионалами. Разработка новых функций при втором подходе выполняется разработчиками приложений (как и в традиционных языках программирования). Полнота использования возможностей встроенных языков (особенно при втором подходе) требует соответствующего уровня квалификации разработчика, если, конечно, в этом есть необходимость. Требования задачи могут быть не столь высокими, чтобы применять всю "мощь" встроенного языка. Во всех языках функции разделяются на группы, часть из которых присутствует практически во всех языках: математические функции, функции работы со строками, SQL-обмен, DDE-обмен и т. д. Каждая из функций во встроенном языке выполняется в синхронном или асинхронном режиме. В синхронном режиме выполнение следующей функции не начинается до тех пор, пока не завершилось исполнение предыдущей. При запуске асинхронной функции управление переходит к следующей функции, не дожидаясь завершения исполнения предыдущей функции. В связи с этим возникает несколько вопросов. С каким приоритетом исполняется каждый из фрагментов, допускается ли рекурсия при обработке событий и если да, то каков уровень вложенности? В SCADA-системах уровень вложенности пока не стандартизован, но оговаривается особо в рамках каждой из них.

SCADA-системы VisiDAQ (Advantech, США) и GeniDAQ включают в себя встроенную среду программирования на языке сценариев [5], совместимом с языком программирования высокого уровня Visual Basic for Applications (VBA). Язык Visual Basic (VB) является одним из наиболее популярных языков программирования. Это позволяет разрабатывать управляющие стратегии практически любого уровня сложности. Редактор сценариев SCADA-систем VisiDAQ и GeniDAQ предназначен для редактирования основного сценария и Basic-сценариев внутри задач. Основной сценарий полностью контролирует процесс выполнения задачи, включая ее запуск и/или останов. Кроме управления задачами, SCADA-система VisiDAQ предоставляет разнообразные команды для ввода-вывода данных, включая операции открытия, чтения, записи, закрытия последовательного порта; операции доступа к информации в центре обработки данных и т. п.

В SCADA-системе Citect [6, 7] встроен гибкий язык программирования Cicode, созданный специально для мониторинга и управления приложениями. Это структурированный язык, похожий на Visual Basic и С. Применение языка Cicode предоставляет пользователю доступ к данным проекта в режиме реального времени, а также ко всем переменным, сигналам тревог, трендам, отчетам и т. д. Язык Cicode поддерживает многозадачность и удаленный вызов процедур, позволяет создавать программы любой степени сложности.

В SCADA-системе InTouch используются скрипты. Скрипты в InTouch — это программные фрагменты, активизируемые по событиям (по нажатию клавиши, кнопки, открытие окна, изменению значения переменной и т. д.).

1.9. Базы данных

Многие SCADA-системы, в частности, Genesis, InTouch используют ANSI SQL синтаксис, который является независимым от типа базы данных. Таким образом, приложения виртуально изолированы, что позволяет менять базу данных без серьезного изменения самой прикладной задачи, создавать независимые программы для анализа информации, использовать уже наработанное программное обеспечение, ориентированное на обработку данных. Так, IndustrialSQL Server компании Wonderware использует подобный подход, впервые превращая реляционную технологию в разумное решение для систем промышленной автоматизации. IndustrialSQL Server — опора пакета промышленной автоматизации Wonderware FactorySuite2000. Несмотря на то, что IndustrialSQL Server поставляется компанией Wonderware как самостоятельный продукт, он, в то же время, является одним из главных компонентов пакета FactorySuite2000, являясь, можно сказать, его "сердцем". Будучи интегрированным со SCADA-системой InTouch, IndustrialSQL Server способен накапливать при помощи серверов ввода/вывода информацию практически от любых измерительных приборов и устройств сбора данных. IndustrialSQL Server представляет собой расширение Microsoft SQL Server. Объединение серверов IndustrialSQL Server и Microsoft SQL Server незаметно для пользователя. Можно сказать, что IndustrialSQL Server превращает Microsoft SQL Server в сервер реляционной базы данных реального времени. При этом клиенты могут напрямую обращаться к IndustrialSQL Server при помощи тех же утилит, что используются сервером Microsoft SQL Server. Выбор Microsoft SQL Server в качестве основы для IndustrialSQL Server объясняется несколькими причинами. Во-первых, в мире существует более 200 миллионов пользователей Microsoft SQL Server. Во-вторых, Microsoft SQL Server является самой продаваемой базой данных для ОС Windows. В-третьих, SQL поддерживается всеми крупными производителями серверов баз данных, большинством средств разработки и языков программирования.

Родственный Citect продукт, называемый Plant2SQL, позволяет предоставлять технологическую информацию, являющуюся прерогативой SCADA-систем. Plant2SQL поддерживает простой доступ к данным технологического процесса, как из приложений, так и со стороны пользователей. Пользователям теперь доступны самые последние данные технологического процесса, что позволяет им принимать решения во всеоружии, полностью владея информацией о процессе производства.

1.10. Организация взаимодействия с устройствами нижнего уровня

Современные SCADA-системы не ограничивают выбор аппаратуры нижнего уровня (контроллеров), так как предоставляют большой набор драйверов или серверов ввода/вывода и имеют хорошо развитые средства создания собственных программных модулей или драйверов новых устройств нижнего уровня. Для подсоединения драйверов ввода/вывода к SCADA-системе в настоящее время используются следующие механизмы:

- ставший стандартом динамический обмен данными (DDE);
- собственные протоколы, связанные со SCADA-системами, реально обеспечивающие самый скоростной обмен данными;
- новый OPC-протокол, который, с одной стороны, является стандартным и поддерживается большинством SCADA-систем (см. приведенную ранее табл. 1.1), а с другой стороны, лишен недостатков протоколов DDE.

Изначально протокол DDE применялся в первых человеко-машинных интерфейсах в качестве механизма разделения данных между прикладными системами и устройствами типа ПЛК (программируемых логических контроллеров). Для преодоления недостатков DDE, прежде всего для повышения надежности и скорости обмена, разработчики предложили свои собственные решения (протоколы), такие как AdvancedDDE или FastDDE — протоколы, связанные с пакетированием информации при обмене с ПЛК и сетевыми контроллерами. Но такие частные решения приводят к ряду проблем:

- для каждой SCADA-системы пишется свой драйвер для поставляемого на рынок оборудования;
- в общем случае, два пакета не могут иметь доступ к одному драйверу в одно и то же время, поскольку каждый из них поддерживает обмен именно со своим драйвером.

Взамен этих протоколов появился новый стандарт OPC (OLE for Process Control), ориентированный на рынок промышленной автоматизации, основанный на COM-технологии фирмы Microsoft. Основная цель OPC-стандарта заключается в определении механизма доступа к данным любого устройства из приложений. OPC позволяет производителям оборудования поставлять программные компоненты, которые стандартным способом обеспечат клиентов данными с ПЛК. При широком распространении OPC-стандарта появляются следующие преимущества:

- OPC позволяет определять на уровне объектов различные системы управления и контроля, работающие в распределенной гетерогенной среде;
- OPC устраняет необходимость использования различного нестандартного оборудования и соответствующих коммуникационных программных драйверов;
- у потребителя появляется больший выбор периферийного оборудования при разработке приложений.

С использованием OPC-решений, интеграция в гетерогенных системах становится достаточно простой. Применительно к SCADA-системам OPC-серверы, расположенные на всех компьютерах системы управления производственного предприятия, стандартным способом могут поставлять данные в программу визуализации, базы данных и т. п., уничтожая, в некотором смысле, само понятие неоднородной системы.

Для обмена данными с контроллерами в SCADA-системе Genie — предшественнике SCADA-системы GeniDAQ, могут использоваться встраиваемые драйверы, DDE-обмен, но в нем не поддерживается OPC-спецификация. SCADA-система GeniDAQ имеет ряд ключевых отличий, обеспечивающих решение более широкого круга задач на новом уровне, в том числе с поддержкой OPC-спецификации.

В SCADA-системе InTouch поддерживается также пакетированный DDE-обмен — FastDDE. Применение последнего заметно повышает эффективность и производительность обмена данными благодаря уменьшению общего количества DDE-пакетов, которыми клиент и сервер обмениваются между собой. Но принципиальные недостатки, связанные с надежностью и зависимостью от количества загруженных в текущий момент приложений Windows, остались. Необходимость в появлении более совершенного технологичного протокола созрела. С целью расширения возможностей стандартного протокола DDE на локальную сеть компания Wonderware предложила протокол NetDDE. Он позволяет приложениям, запущенным на объединенных в локальную сеть компьютерах, вести DDE-обмен. Для реализации функций OPC-клиента Wonderware предлагает OPCLink-сервер, преобразующий OPC в SuiteLink-протокол. Параметры производительности протокола SuiteLink превосходят параметры DCOM. OPCLink-сервер обеспечивает прием информации с OPC-сервера и передачу ее по протоколу SuiteLink в SCADA-систему InTouch и наоборот. Именно OPCLink-сервер рекомендуется устанавливать на одном узле с OPC-сервером, чтобы для сетевых передач использовался SuiteLink-протокол, а не DCOM.

Для обмена данными с контроллерами в SCADA-системе Citect могут использоваться следующие способы: встраиваемые драйверы, DDE-обмен, OPC-протоколы. Первый путь предполагает создание динамических библиотек, выполняющих функцию драйверов. В составе SCADA-системы Citect поставляется более чем 120 драйверов ввода/вывода. Связь через DDE-сервер использует стандартный коммуникационный протокол Windows. SCADA-система Citect поддерживает связь с любым DDE-сервером. Система Citect может функционировать в качестве и OPC-сервера и OPC-клиента.

1.11. Открытость SCADA-систем

Система является открытой, если для нее определены и описаны используемые форматы данных и процедурный интерфейс, что позволяет подключить к ней "внешние", независимо разработанные компоненты.

Разработка собственных программных модулей. Перед фирмами-разработчиками систем автоматизации часто встает вопрос о создании собственных (не

предусмотренных в рамках SCADA-систем) программных модулей и включение их в создаваемую систему автоматизации. Поэтому вопрос об открытости системы является важной характеристикой SCADA-систем. Фактически открытость системы означает доступность спецификаций системных (в смысле SCADA) вызовов, реализующих тот или иной системный сервис. Это может быть и доступ к графическим функциям, функциям работы с базами данных и т. д.

Драйверы ввода-вывода. Современные SCADA-системы не ограничивают выбор аппаратуры нижнего уровня, так как предоставляют большой набор драйверов или серверов ввода-вывода и имеют хорошо развитые средства создания собственных программных модулей или драйверов новых устройств нижнего уровня. Сами драйверы разрабатываются с использованием стандартных языков программирования. Вопрос, однако, в том, достаточно ли только спецификаций доступа к ядру системы, предоставляемых фирмой-разработчиком в штатном комплекте (система Trace Mode), или для создания драйверов необходимы специальные пакеты (системы FactoryLink, InTouch), или же, вообще, разработку драйвера нужно заказывать у фирмы-разработчика.

Разработки третьих фирм. Объекты ActiveX — это объекты, в основе которых лежит Microsoft COM. Технология COM определяет общую схему взаимодействия компонентов программного обеспечения в среде Windows и предоставляет стандартную инфраструктуру, позволяющую объектам обмениваться данными и функциями между прикладными программами. Большинство SCADA-систем являются контейнерами, которые уведомляются ActiveX о происшедших событиях. Этот факт очень важно оценивать при выборе SCADA-пакета, поскольку это расширяет область применения системы непрофессиональными программистами.

Подавляющее большинство SCADA-пакетов, несмотря на их реализацию на стандартной платформе Windows, являются закрытыми, если речь идет о структуре данных, с которыми они работают. Конечно большинство из них (в том числе GeniDAQ) имеют стандартные интерфейсы типа DDE, OLE, ODBC и др., сохраняя при этом специализированные, нестандартные форматы файлов для архивов, графиков и сигнализаторов. Для того чтобы получить доступ к данным в этих системах, необходимо воспользоваться специализированными пакетами разработчика, а часто это вообще невозможно. SCADA-система Citect не делает секрета из формата своих данных — все параметры и данные приложения хранятся в формате dBase, понимаемом любым пакетом. Кроме этого, данные могут сохраняться и в более современных SQL базах данных, используя стандартные интерфейсы Citect. Эта открытость будет по достоинству оценена теми, кто не имел возможности, скажем, отобразить на экране данные из архива в том виде, котором требуется оператору, а не так, как умеет SCADA.

1.12. О реальном времени

Одним из существенных недостатков SCADA-систем на платформах Windows 95/98/Me по сравнению с таковыми же системами на платформах ОС реального времени (ОС РВ) является отсутствие поддержки реального времени.

Ситуация резко изменилась с появлением Windows NT/2000/XP. Выход в свет этой ОС стимулировал разработку новых подходов в поддержке жесткого реального времени. Прежде всего, сама по себе Windows NT весьма успешно теснит ОС РВ. Тем не менее, Windows NT имеет ряд ограничений. Такие ее особенности, как предпочтение аппаратного прерывания над программным, выполнение в подпрограмме обработки аппаратных прерываний лишь необходимых действий с выполнением последующей обработки через очередь отложенных процедур, отсутствие приоритетной обработки процессов в очереди отложенных процедур, не позволяют отнести Windows NT к категории классических ОС реального времени. Ряд фирм (LP Elektronik, Imagination Systems, RadSys, Spectron Microsystems, Ventur Com) предприняли более радикальные попытки превратить Windows NT в ОС жесткого реального времени. Рассмотрим некоторые ключевые особенности реализации такой идеи на подсистеме реального времени RTX (Real Time Extension), предложенной фирмой Ventur Com. Именно эта реализация получает в настоящее время наиболее широкое распространение. Фирмы-разработчики SCADA-систем незамедлительно начали предлагать применение новых решений. Так, набор прикладных интерфейсов программирования RTX 4.1 (Ventur Com) в SCADA-системе FIX позволяет:

- осуществлять полный контроль над задачами реального времени;
- использовать фиксированную систему из 128 приоритетов для контроля RTX-задач;
- применять стандартные средства обмена данными между задачами;
- обращаться к стандартным функциям из Win32API.

Появление подобных решений наряду с собственными характеристиками Windows NT наносит сильный "удар" по SCADA-системам на базе ОС РВ, поскольку отнимает у них очень важный "козырь" — преимущества жесткого реального времени, и, для некоторых приложений, теснит применение ОС РВ во встраиваемых системах. Сегодня для весьма широкого спектра промышленных приложений уже есть реальная возможность использовать Windows NT как единую операционную систему, со всеми вытекающими отсюда последствиями, работающую даже в бездисковых конфигурациях в сетевых контроллерах ввода/вывода, скажем на платформе CompactPCI или VME, наряду с использованием на рабочем месте оператора-технолога.

Не секрет, что во многих SCADA-пакетах обновление данных организовано на основе опроса параметров в рамках одной основной задачи и соответствующего обновления экрана по заданным промежуткам времени. Этот алгоритм очень напоминает корпоративную многозадачность, реализованную в Windows 3.X. При этом последствия для работы всего приложения практически идентичны — если по каким либо причинам работа любой функции основной задачи нарушается, то нарушается работа всего пакета. Так, например, задержка в получении данных, необходимых для обновления результатов на одном(!) из графиков часто приводит в таких системах к полной остановке работы всего приложения до тех пор, пока запрос не будет обработан. Хотя здесь нечему удивляться. Многие SCADA-пакеты появились в эпоху Windows 3.X и их ядро не изменялось с тех далеких пор.

Ядро Citect было изначально задумано и реализовано с учетом принципов построения систем реального времени. Анимация объектов на экране, обмен данными, функции, созданные пользователем и встроенные в систему обрабатываются своими независимыми потоками с четким распределением приоритетов. При этом многопоточность реализована по схеме с вытеснением, когда задача с большим приоритетом вытесняет задачу с меньшим, а внутри класса задач с одним приоритетом вытеснение производится по круговой схеме (round robin) путем выделения квантов процессорного времени. Для искушенных пользователей предусмотрена возможность непосредственного использования объектов синхронизации(!) работы потоков и управления их созданием и завершением.

Что думает оператор про SCADA-систему, когда он нажимает на кнопку включения мотора (любого ответственного механизма), но ничего не происходит? А если сигнал тревоги (alarm), сгенерированный на объекте, поступает к оператору не сразу, а через 5-10 секунд? Адекватная работа с сигналами из реальной аппаратуры, особенно когда они составляют замкнутую цепь управления — задача номер 1 для любого SCADA-пакета. К сожалению, большинство SCADA-пакетов очень плохо масштабируются при изменении нагрузки, т. е. при изменении количества сигналов и параметров (окон, элементов управления, количества выполняемых функций и т. д.). Если мы будем сравнивать скорость отклика системы при минимальном числе входных параметров, то получим весьма схожие результаты, но принципиально различные при увеличении нагрузки. Основной причиной этого является отсутствие в большинстве SCADA-пакетов ядра реального времени, так как это сделано в Citect. Именно поэтому теоретическим и практическим пределом для обычных SCADA-пакетов является 10000-30000 параметров (tags), в то время как Citect может работать с числом переменных до 500000. И эта величина не является теоретическим пределом, скорее практическим. Конечно, такие большие системы реализуются достаточно редко, но даже и в диапазоне 5000-10000 параметров Citect имеет бесспорные и ощутимые преимущества. SCADA-система Citect работает с разрешением по времени, равным одной миллисекунде. Это относится как к работе с данными, так и к обновлению графиков реального времени (trends) и других объектов.

1.13. Средства визуализации

Средства визуализации — одно из базовых свойств SCADA-систем. В каждой из них существует графический объектно-ориентированный редактор с определенным набором анимационных функций. Используемая векторная графика дает возможность осуществлять широкий круг операций над выбранным объектом. Объекты могут быть простыми (линии, прямоугольники, текстовые объекты и т. д.) и сложными. Возможности агрегирования сложных объектов в разных SCADA-системах различны. Все SCADA-системы включают библиотеки стандартных графических символов, библиотеки сложных графических объектов, обладают целым рядом других стандартных возможностей. Но, тем не менее, каждая SCADA-система по-своему уникальна и, несмотря на поддержание стандартных функций, обладает присущими только ей особенностями. При рассмотрении графических возможностей

SCADA-систем InTouch и Citect предлагается обратить внимание не только на возможности инструментов по созданию графических объектов, но и на другие предоставляемые пользователю услуги, облегчающие и ускоряющие процесс разработки приложений.

1.14. Состояние тревоги и события

Состояние тревоги, в дальнейшем Alarm — это некоторое сообщение, предупреждающее оператора о возникновении определенной ситуации, которая может привести к серьезным последствиям, и потому требующее его внимания, а часто и вмешательства. А принял ли оператор сообщение об Alarm? Чтобы снять эти сомнения, в системах управления принято различать неподтвержденные и подтвержденные Alarm. Alarm называется подтвержденным после того, как оператор отреагировал на сообщение об Alarm. До этого Alarm оставался в состоянии неподтвержденного. Наряду с Alarm в SCADA-системах существует понятие события. События представляют собой обычные статусные сообщения системы и не требуют реакции оператора. Обычно событие генерируется при возникновении в системе определенных условий (типа регистрации оператора в системе). От эффективности подсистемы Alarm зависит скорость идентификации неисправности, возникшей в системе, или идентификации технологического параметра, вышедшего за установленные регламентом границы. Быстродействие и надежность этой подсистемы могут существенно сократить время простоя технологического оборудования. Например, если оператор не получит вовремя информацию о том, что двигатель насоса перегрелся, это может привести в лучшем случае к выходу насоса из строя, а то и к крупной аварии. Причины, вызывающие состояние Alarm, могут быть самыми разными. Неисправность может возникнуть в самой SCADA-системе, в контроллерах, каналах связи, в технологическом оборудовании. Может выйти из строя датчик или нарушатся его метрологические характеристики. Параметры технологического процесса могут выйти за границы, установленные регламентом и т. д.

1.15. Отображение и архивирование данных

Графическое представление значений технологических параметров во времени способствует лучшему пониманию динамики технологического процесса предприятия. Поэтому подсистема создания трендов (графиков) и хранения информации о параметрах с целью ее дальнейшего анализа и использования для управления является неотъемлемой частью любой SCADA-системы. Тренды реального времени отображают динамические изменения параметра в текущем времени. При появлении нового значения параметра в окне тренда происходит прокрутка графика справа налево. Таким образом, текущее значение параметра выводится всегда в правой части окна. Тренды становятся архивными после того, как данные будут записаны на диск, и можно будет использовать режим прокрутки предыдущих значений назад с целью посмотреть прошлые значения. Отображаемые

данные тренда в таком режиме будут неподвижны, и будут отображаться только за определенный период.

1.16. Стоимость системы

Стоимость освоения системы. Процедура освоения SCADA-систем достаточно проста с точки зрения программиста и не требует длительного времени, поэтому эти затраты относительно невелики. Основной составляющей стоимости является оплата труда программистов, осуществляющих эту работу.

Стоимость сопровождения или "стоимость владения". Эта составляющая обычно наиболее "скрыта от глаз покупателя" и зависит от многих факторов. Например:

- стоимость "риска" покупки, который определяется такими параметрами как рыночная надёжность фирмы-дистрибутора инструментального пакета (трудно говорить о надёжности фирмы, если её, скажем, штат 1-5 человек), рыночная стабильность фирмы-изготовителя продукта;
- стоимость коммуникаций с фирмой-поставщиком;
- "время реакции" поставщика на проблемы покупателя;
- наличие реального прикладного опыта и хорошего знания поставляемого продукта специалистами фирмы-поставщика, наличие в принципе у поставщика специалистов по продукту;
- степень открытости, адаптируемости и модернизируемости продукта.

Эти и многие другие факторы, влияющие на "стоимость владения" необходимо учитывать при выборе системы. Можно подчеркнуть, что концентрация разработчиков SCADA-систем на поле Windows NT способствует снижению "стоимости владения" пользователем этими продуктами.

Стоимость разработки прикладных систем. Стоимость, связанная с трудозатратами на разработку прикладных программ при использовании SCADA-систем, существенно уменьшается по сравнению с использованием традиционного программирования. В качестве примера можно привести работы по созданию системы информационного обслуживания в CERN, которая содержала несколько десятков узлов. Эти работы вместе с отладкой заняли около трех месяцев и выполнялись с помощью системы FactoryLink. Но следует учесть, что вышесказанное относилось к стоимости разработки системы. Необходимо отметить, что стоимость изготовления системы составляет обычно 30-50% от стоимости разработки системы.

Срок окупаемости SCADA-системы. Чтобы оценить время окупаемости SCADA-системы необходимо учесть множество факторов, включая количество проектов, реализуемых на основе этой системы, стоимость этих проектов и т. д. Ориентировочно, если речь идет о бизнесе системного интегратора, реализация 2-3 проектов при приобретении системы разработки SCADA окупает ее. Одним из важных свойств SCADA всегда была открытость. Сейчас открытость дополняется новыми средствами передачи данных между процессами — OLE, OPC, встраиваемыми программными объектами (ActiveX).

1.17. SCADA-продукты на российском рынке

В [8] приведены характеристики SCADA-систем, представленных на российском рынке. Так как эти характеристики получены в результате анкетирования компаний — поставщиков SCADA-продуктов, то они не претендуют на исчерпывающий перечень SCADA-систем. Тем не менее, эти характеристики являются весьма информативными для выбора SCADA-системы.

Анкета, использованная при получении указанных сведений, составлена с учетом мнения ряда компаний о том, какие характеристики SCADA-систем важны при их оценке и выборе. Анкета включает около 100 критериев, распределенных по *четырем категориям и основным группам* внутри каждой категории. Категории перечислены в соответствии с их *приоритетностью* (с точки зрения выбора SCADA-системы) и соответствуют основным этапам жизненного цикла SCADA-систем.

1.17.1. Базовые свойства

Базовые свойства SCADA-системы определяют идеологическую и функциональную основу АСУ ТП.

- ❑ *Открытость*: интерфейс прикладных программ API (Application Program Interface) к базе данных реального времени; стандартный интерфейс ODBC к реляционным БД и архивам; управление программируемыми логическими контроллерами PLC через DDE, DLL, OLE и OPC; обмен с приложениями через API, DLL, DDE, OLE, COM; API-доступ к архивам и др.
- ❑ *Поддержка контроллеров*: количество поддерживаемых контроллеров; наличие промышленных шин Profibus, Canbus, Foundation fieldbus, Modbus и др.; связь с контроллерами напрямую, по собственному протоколу, по промышленной шине или протоколу третьей фирмы; доступ к контроллерам — опрос, получение меток времени, инициативная передача, подписка; наличие средств разработки драйверов; среднее время разработки нового драйвера и др.
- ❑ *Архивация и тренды*: вывод в тренды в РВ; запоминание информации во временном хранилище — буфере, файле — для последующей перезаписи в архив; непосредственный вывод в архивные тренды; возможность просмотра трендов "через лупу"; максимальное число кривых в тренде; максимальное число трендов в окне; выбор и/или отслеживание параметров с использованием предустановки или онлайн-овой; одновременный вывод трендов в РВ и архивных трендов; возможность вывода значения параметра в точке, указанной курсором; использование для работы с трендами модулей или объектов ActiveX и др.
- ❑ *Алармы*: количество уровней приоритетов; возможность группирования алармов; онлайн-овая фильтрация; маскирование алармов по нижним/верхним границам изменения параметров; наличие алармов по скорости изменения параметров или по отклонению параметров от уставки; автоматическое выполнение определенных действий при возникновении оговоренных алармов; генерация E-mail сообщений и др.

- Возможности HMI:* наличие объектно-ориентированного графического редактора; библиотека стандартных графических символов; техника "drag-and-drop"; библиотека сложных графических объектов; стандартные возможности оконного редактирования — масштабирование, изменение размеров, прокрутка, манипуляции с "пером" и т. п.; возможность работы с трендами; возможность навигации по страницам изображения; анимация, мультимедиа и пр.
- WEB-технологии:* Java, ActiveX; мониторинг; управление и пр.
- Система отчетов:* встроенный генератор отчетов; метод "вырезать/вставить"; печать, архивация отчетов и др.
- Масштабируемость:* по базе данных РВ; по количеству узлов в сети.
- Опции/автоматизация:* управление технологическим процессом на основе статистической обработки полученной информации; сохранение конкретной конфигурации системы в файле с последующей ее перезагрузкой; планирование заданий; автообработка событий при изменении параметра и др.
- Поддержка ОС:* Windows 95, Windows NT, AIX, UNIX, VMS, QNX, DOS.
- Защита доступа:* через пароль; собственная система защиты; уровни доступа — чтение, запись, приоритеты, классы, авторизация.
- Резервирование:* узла, сервера ввода/вывода, трендов, алармов, контроль резервирования.
- Стоимость.*
- Рыночная политика:* ценовая и лицензионная гибкость.

1.17.2. Возможности среды разработки

Возможности среды разработки SCADA-системы определяется общими средствами расширения инструментальных возможностей SCADA-пакетов.

- Разработка в on-line.
- Тестирование/отладка в on-line.
- Возможность симуляции.
- Разработка распределенных приложений.
- Язык программирования/сценариев.
- Графические языки.
- Возможность одновременной разработки нескольких проектов.
- Многопользовательская разработка.
- Наличие пакетов разработчика (toolkit), позволяющих создавать драйверы ввода-вывода, графические объекты и т. п.
- Пакет комплексной автоматизации. Наличие пакета комплексной автоматизации означает вхождение SCADA-системы в интегрированную программную систему, предназначенную для автоматизации управления в единой информационной

среде на основных производственных уровнях – от уровня непосредственного управления (уровень контроллеров) до уровня автоматизированной системы управления предприятием (АСУП).

1.17.3. Возможности конфигурирования системы

- Может ли любой узел быть сервером ввода-вывода, сервером трендов, сервером алармов.*
- Может ли любой узел быть клиентом, взаимодействующим с любым сервером.*
- Может ли любой узел быть только мониторинговой подсистемой.*
- Конфигурируемость по числу точек ввода-вывода.*
- Максимальное число точек ввода-вывода.*

1.17.4. Характеристики сопровождения/эксплуатации

Характеристики сопровождения/эксплуатации SCADA-систем определяют эксплуатационные возможности и услуги пользователям при сопровождении систем распространителями.

- Локализация и мультиязыковая поддержка.
- Рыночный стаж.
- Простота использования.
- Время обучения технического персонала.
- Наличие документации на русском языке.
- Наличие продуктов других фирм, поддерживающих данный SCADA-пакет.
- Количество инсталляций.
- Наличие обучающих курсов.
- Время изучения продукта.

Перечень SCADA-систем на российском рынке, фирм-производителей и характеристик SCADA-систем по состоянию на 1999 год имеется в [8]. Еще раз обращаем внимание на то, что эти данные не являются исчерпывающими, так как получены путем анкетирования фирм-производителей. При этом анкетировались не все фирмы (не все они были известны при анкетировании) и не все фирмы откликнулись на просьбу заполнить анкеты. К тому же, результаты анкетирования к настоящему времени несколько устарели. Тем не менее, эти данные очень важны и существенны — их изучение принесет проектировщику большую пользу.

1.17. Заключение

По функциональным возможностям все рассмотренные SCADA-системы в целом сравнимы. Технология программирования близка к интуитивному восприятию автоматизируемого процесса. Мощное объектно-ориентированное

программирование, используемое в большинстве этих пакетов, делает эти продукты легкими в освоении и доступным для широкого круга пользователей. Все системы можно считать в той или иной степени открытыми, обеспечивающими возможность дополнения функциями собственной разработки, имеющими открытый протокол для разработки собственных драйверов, развитую сетевую поддержку, возможность включения ActiveX-объектов и доступ к стандартным базам данных. Важной особенностью всех SCADA-систем является поддержка OPC-спецификации. Это делает их привлекательными. Построение прикладной системы на основе любой из рассмотренных SCADA-систем резко сокращает набор необходимых знаний в области классического программирования, позволяя концентрировать усилия по освоению знаний в прикладной области. У разработчиков SCADA-систем на платформе Windows NT/2K/XP появилась возможность использовать расширение реального времени (RTX). Следует отметить тенденции включения SCADA-систем в системы комплексной автоматизации предприятия. Это обеспечивает точную, своевременную информацию на каждом уровне производства. Применение в SCADA-системах новых технологий, разработка инструментальных средств комплексной автоматизации предприятия свидетельствуют о стремлении и возможности фирм-разработчиков постоянно совершенствовать свои продукты, что является немаловажным фактором при выборе инструментального средства, даже если не все его технологические решения в ближайшее время будут использованы Вами. Так как общее поле деятельности ведущих компаний-производителей описываемых инструментальных систем сегодня концентрируется в области MS Windows, а общие технические возможности систем достаточно близки, то главный упор делается на качество технической поддержки, на качество обучения пользователей, на концентрацию и качество дополнительных комплексных услуг по освоению и внедрению конечной системы управления. Другими словами, упор делается на сокращение издержек системных интеграторов и конечных пользователей на инжиниринг и менеджмент своих проектов, на уменьшение стоимости сопровождения конечной системы. Именно эти показатели сегодня, в основном, влияют на рейтинг и рыночный успех той или иной SCADA-системы. Пожалуй, эти показатели даже более важны, чем абсолютные стоимостные характеристики SCADA-систем.

На основе сравнительного анализа SCADA-систем можно выделить следующую представительную группу SCADA-систем:

- Intouch (одна из наиболее мощных SCADA-систем);
- Citect (одна из наиболее мощных SCADA-систем);
- TRACE MODE (SCADA-система среднего класса отечественной разработки и производства);
- GeniDAQ (простая и недорогая SCADA-система).

SCADA-система Intouch (компания Wonderware, США) насчитывает более 100000 инсталляций во всем мире, причем 3000 из них выполнены компанией KLINKMANN (компания обеспечивает техническую поддержку и документацию на русском языке). В KLINKMANN разработано более 100 драйверов промышленного оборудования,

включая систему для беспроводной связи и управления на основе стандарта GSM, для применения с программным обеспечением Wonderware.

SCADA-система Vijeo Citect (компания Citect Pty. Ltd, Австралия) занимает более 6% мирового рынка SCADA-систем и более 60% австралийского рынка. Эта система в наибольшей степени отвечает свойству открытости в возможности подключения практически к любым контроллерам. Количество подключаемых точек ввода-вывода больше, чем в SCADA-системе Intouch, но Vijeo Citect проще в освоении. На российском рынке обеспечивается техническая поддержка Vijeo Citect, имеется система центров обучения и документация на русском языке. Демонстрационная версия Vijeo Citect имеет ограничения, несущественные для использования в учебном процессе (шесть часов непрерывной работы, при работе с внешними каналами ввода-вывода — 10 минут).

SCADA-система TRACE MODE (компания AdAstra Research Group, Ltd, Москва) представляет собой 32-разрядную SCADA и SoftLogic-систему для разработки АСУ ТП мирового класса. С более чем 3500 инсталляциями TRACE MODE является одной из наиболее покупаемых в России для разработки крупных диспетчерских комплексов и программирования PC-контроллеров. Система поддерживает более 300 марок контроллеров и УСО. Благодаря новым технологиям сквозного программирования операторских станций и контроллеров, а также автоматического построения проекта, мощь TRACE MODE теперь сочетается с простотой разработки. TRACE MODE имеет самую современную архитектуру, основанную на DCOM. Поддерживаются технологии OPC (сервер/клиент), ActiveX, Internet/Intranet, SQL/ODBC, DDE.

SCADA-система GeniDAQ является самой простой и недорогой SCADA-системой, наиболее простой в изучении и освоении. Ее демонстрационная версия имеет ограничения, несущественные для использования в учебном процессе (два часа непрерывной работы, ограниченное число каналов ввода-вывода и др.). С другой стороны, *GeniDAQ* является вполне представительной SCADA-системой.

Из сказанного следует, что для применения в учебном процессе перспективным является изучение SCADA-систем GeniDAQ и Vijeo Citect, причем именно в такой последовательности.

Глава 2. OPC — промышленный стандарт и средство интеграции компонентов в промышленной автоматизации

Ранее указывалось, что информационная автоматизированная система управления промышленным предприятием имеет открытую архитектуру (рис. 2.1), включающую следующие неотъемлемые уровни [1]:

- Нижний уровень, включающий полевые шины и отдельные контроллеры (Field Management), представляющие собой "интеллектуальные" технологические устройства: датчики, контроллеры, механизмы и т. д. Поток информации от нижнего уровня должен быть предоставлен пользователю и всем приложениям верхнего уровня, использующим их, посредством цифровых коммуникационных протоколов связи. При этом в системе не должно возникать проблем несовместимости.
- Уровень АСУ ТП (Process Management) — уровень работы SCADA-систем для сбора данных и диспетчерского управления технологическими процессами на производстве. Этот уровень обеспечивает вторичную обработку данных, которые получены с нижнего уровня, сохранение данных и их доступность приложениям и пользователям верхнего уровня.
- Уровень автоматизированной системы управления предприятием (АСУП) — уровень приложений управления ресурсами предприятия (Business Management). Информация с уровня АСУ ТП должна быть доступной для уровня АСУП, т. е. доступ к данной информации со стороны прикладных программ также не должен вызывать проблемы несовместимости.

Для обеспечения совместимости между уровнями и создания эффективной интегрированной системы управления предприятием системный интегратор или разработчик АСУ ТП должен извлекать данные технологического процесса в реальном времени с самого нижнего уровня и обеспечивать "прозрачный" путь получаемым данным к самым верхним уровням. Чтобы получить систему, отвечающую всем потребностям заказчика, системному интегратору или разработчику необходимо использовать инструментальные средства управления различных уровней — SCADA-пакеты, базы данных, электронные таблицы. Ключ к этому — открытая и эффективная коммуникационная архитектура взаимодействия между приложениями, которую предлагает стандарт OPC (OLE for Process Control).

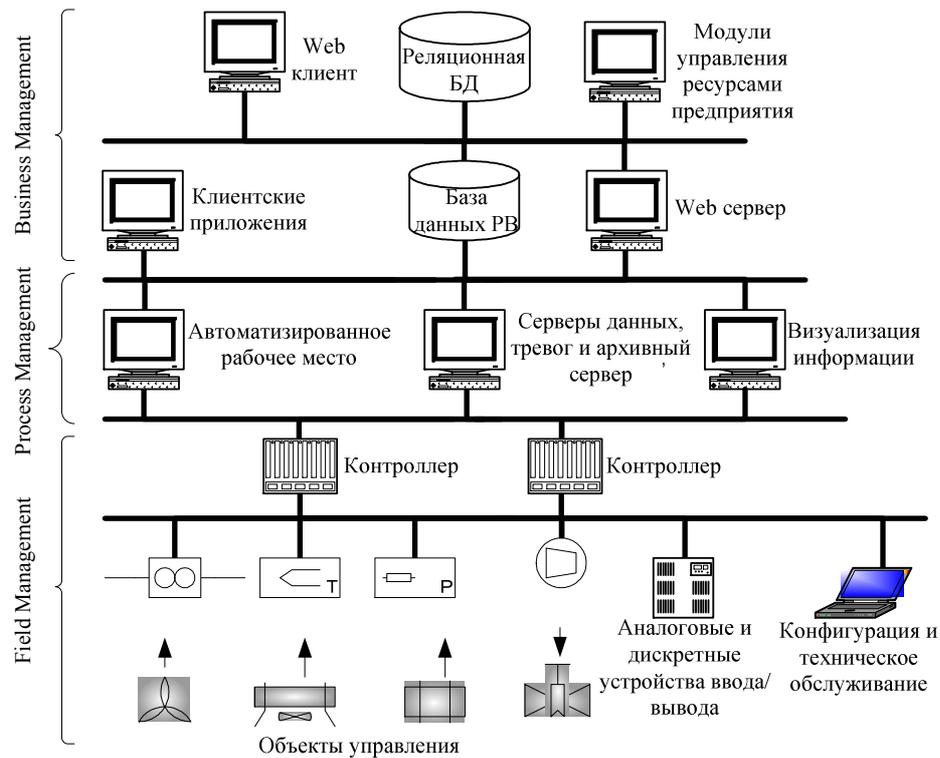


Рис. 2.1. Архитектура информационной автоматизированной системы управления промышленным предприятием

2.1. Что такое OPC?

Попытаемся определить, что такое OPC. *OPC* — технология связывания и внедрения объектов для систем промышленной автоматизации, предназначенная для обеспечения универсального механизма обмена данными между датчиками, исполнительными механизмами, контроллерами, устройствами связи с объектом и системами представления технологической информации, оперативного диспетчерского управления, а также системами управления базами данных.

На сегодняшний день технология OPC в определенной степени реализована, но продолжает развиваться. Консорциум OPC Foundation, объединяющий более 300 производителей средств промышленной автоматизации, разрабатывает все аспекты, связанные с взаимодействием между компонентами программного обеспечения, между программным обеспечением и между системами типа SCADA и технологическим оборудованием. OPC Foundation привлекает к разработке спецификаций обмена ведущих производителей оборудования и систем автоматизации, которые стараются максимально учесть свой опыт и предоставить абсолютно всё необходимое тому, кто будет использовать OPC. Существуют много

спецификаций OPC — Data Access (доступ к данным реального времени), Alarms & Events (обработка тревог и событий), Historical Data Access (доступ к архивным данным) и т. д. Поэтому можно определить OPC как стандарт взаимодействия между программными компонентами сбора данных и управления, основанный на объектной модели COM/DCOM фирмы Microsoft. Через интерфейсы OPC одни приложения могут читать или записывать данные в другие приложения, обмениваться событиями, оповещать друг друга о нештатных ситуациях, осуществлять доступ к данным, зарегистрированным в архивах. Эти приложения могут располагаться как на одном компьютере, так и быть распределенными в сети. При этом, независимо от фирмы-поставщика, стандарт OPC, признанный и поддерживаемый всеми ведущими фирмами-производителями SCADA-систем и оборудования, обеспечит их совместное функционирование.

Популярный класс OPC-приложений представляют собой OPC-серверы конкретных аппаратных устройств, обеспечивающие предоставление информации о состоянии параметров многочисленных устройств технологического процесса, полученной OPC-приложением, OPC-клиентам на локальном компьютере или в компьютерной сети. OPC-сервер создает свою абстракцию устройств (это загружаемый модуль, предоставляющий интерфейс нижнего уровня с оборудованием, на котором исполняется OPC-сервер). OPC-сервер скрывает аппаратно-зависимые детали, такие как интерфейс ввода/вывода, контролеры прерываний и механизм коммуникации в многопроцессорных системах — любые функции, зависящие от конкретной архитектуры и машин, позволяя любому OPC-клиенту записывать и считывать данные с устройства. Устройство, для которого есть OPC-сервер, может использоваться вместе с любой современной SCADA-системой.

Современные SCADA-системы не ограничивают выбор аппаратных средств нижнего уровня, так как SCADA-пакеты предоставляют большой набор драйверов или серверов ввода/вывода (в т. ч. OPC-серверы) и имеют хорошо развитые средства создания собственных программных модулей или драйверов для новых и нестандартных устройств нижнего уровня. Производители аппаратных средств нижнего уровня, используя спецификации OPC, имеют возможность разрабатывать свой OPC-сервер для обеспечения доступа к данным реального времени и передачи данных приложениям — клиентам различных производителей программного обеспечения для промышленной автоматизации (в т. ч. SCADA-системам).

2.2. DCOM и OPC-приложения

Технология OPC основана на модели распределенных компонентных объектов *DCOM* — Distributed COM. Модель представляет собой сетевое расширение COM или распределённую COM (Component Object Model). Технология COM — Моделей Компонентных Объектов — создана первоначально Microsoft для совместного использования различных офисных приложений в Windows, например, электронных таблиц Excel, редактора Word и т. п.

COM поддерживает модель "клиент-сервер" [9]. Объекты, называемые серверами, предоставляют некие функции в распоряжение объектов, называемых клиентами.

Серверы всегда являются COM-объектами, т. е. объектами, которые подчиняются спецификации COM. С другой стороны, клиенты могут быть COM-объектами или не быть таковыми. Это значит, что некоторые объекты могут быть простыми объектами C++, приложениями Visual Basic и т. п. Каждый COM-объект существует внутри конкретного сервера. Этот сервер содержит программный код реализации операций, а также данные активного COM-объекта. Один сервер может обеспечивать несколько объектов и даже несколько COM-классов. Используются три типа серверов (рис. 2.2):

- ❑ внутризадачный сервер (in-process) — объекты находятся в динамически подключаемой библиотеке и, следовательно, выполняются в том же процессе, что и клиент;
- ❑ локальный сервер (local) — объекты находятся в отдельном процессе, выполняются в том же компьютере, что и клиент;
- ❑ удаленный сервер (remote) — объекты находятся в DLL или в отдельном процессе, который расположен на удаленном от клиента компьютере.

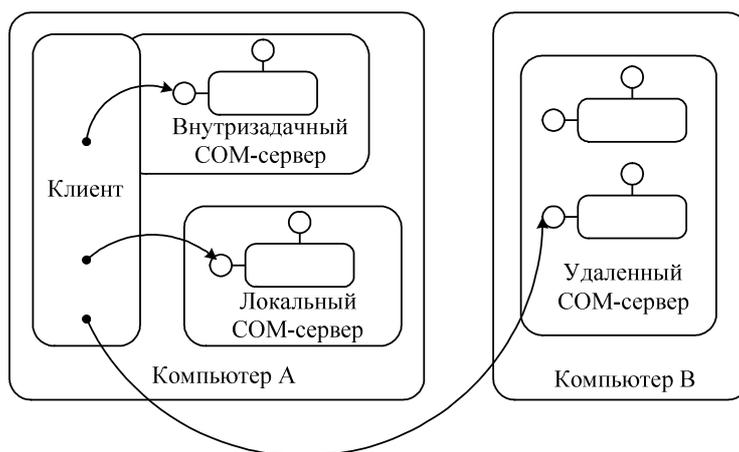


Рис. 2.2. Разновидности OPC-серверов

Клиенты и локальные серверы находятся в различных адресных пространствах. OPC берет на себя заботу о передаче данных между ними. В случае внутризадачных серверов, родительский процесс загружает DLL-файл, содержащий COM-сервер. Это означает, что внутризадачный сервер находится в том же адресном пространстве, что и вызвавший его процесс. В случае DCOM вызов любой функции объекта перехватывается специальным агентом-посредником, называемым Proxy Manager или Stub Manager. Proxy Manager получает все запросы на транспортировку и определяет, стандартные ли данные запрашивает или предлагает клиент. Если это так, Proxy Manager передает запрос подходящему Proxy-объекту. Подходящие Proxy и Stub — это объекты, предназначенные для транспортировки данного интерфейса и/или типа данных. Аналогичный процесс происходит на стороне Stub Manager. Встроенные Proxy и Stub называются транспортировщиками интерфейса.

Популярным вариантом OPC-клиента является SCADA-система. С помощью OPC-серверов SCADA-система может обращаться к любому аппаратному устройству (рис. 2.3) путем вызова определенных функций OPC-сервера, подписываться на получение определенных данных какого-либо канала или устройства с требуемой частотой. В свою очередь, OPC-сервер опрашивает аппаратное устройство, вызывает известные функции клиента, подтверждает клиенту получение данных и посылает требуемые клиенту данные.

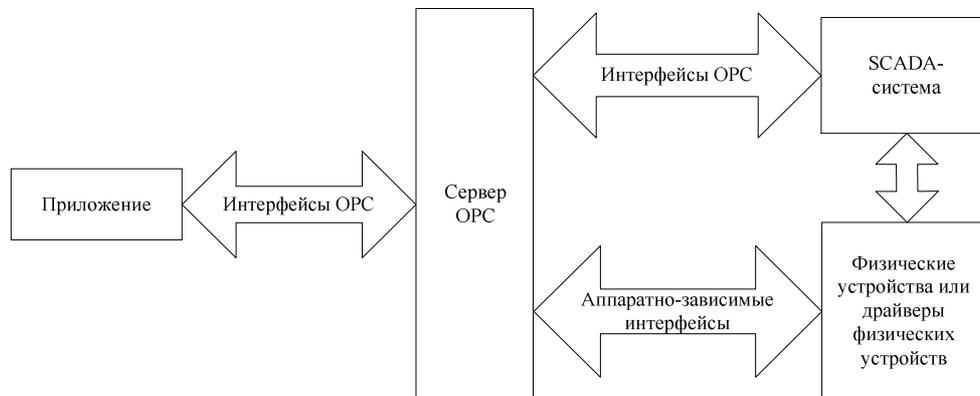


Рис. 2.3

Обмен данными между OPC-клиентом и OPC-сервером может быть реализован в трех режимах — синхронного чтения/записи, асинхронного чтения/записи и режиме подписки (только чтение).

При синхронном чтении/записи клиент посылает серверу запрос со списком интересующих его переменных и ждёт, когда сервер его выполнит.

При асинхронном чтении/записи клиент посылает серверу запрос, а сам продолжает работать. Когда сервер выполнил запрос, клиент получает уведомление. Этот режим обладает большей гибкостью и рекомендуется в большинстве случаев, т. к. он минимизирует затраты процессорного времени и сетевых ресурсов, особенно при передаче большого количества данных.

В случае *подписки* клиент передаёт серверу список интересующих его переменных, а сервер присылает клиенту информацию об изменившихся переменных из этого списка. При этом сервер будет передавать клиенту информацию только тогда, когда данные изменились, причём эти данные передаются единым блоком. Эти меры позволяют существенно ускорить обмен данными, особенно если речь идёт о взаимодействии через сеть.

Кроме механизмов обмена данными, при реализации OPC-серверов необходимо учитывать еще один важный аспект — источник получаемых данных. В зависимости от реализации OPC-сервера, клиенты могут свободно получать данные от разных источников — непосредственно от устройства или из кэша OPC-сервера.

2.3. Почему OPC?

Ранее уже упоминалось, что основная цель OPC-стандарта заключается в определении механизма доступа к данным любого устройства из различных приложений. OPC позволяет производителям оборудования стандартизировать свои продукты, обеспечивает подключение оборудования различных производителей, в т. ч. подключение к SCADA-системам.

Идеальной выглядела бы следующая картина. Все в мире признают OPC своим стандартом. При этом все поставщики оборудования снабжают свои продукты OPC-серверами. Все поставщики программ для систем управления делают свои продукты OPC-клиентами — а все производители операционных систем поддерживают в своих ОС технологии COM/DCOM, а также предоставляют сервисный инструментарий. И при этом все это делают на высоком профессиональном уровне и очень грамотно рассказывают сборщикам систем, как это всё собирать и конфигурировать. Тогда пользователи будут иметь широкую возможность выбора оптимальных для своей системы компонентов, а не создавать систему управления "с нуля" и не выбрасывать свой имеющийся контроллер. Таким образом, по образному выражению Теркеля [10], OPC — это свобода выбора.

Преимущество применения стандарта OPC с точки зрения интеграции достаточно прозрачно. С применением OPC-стандарта появляется возможность интегрировать в единую систему неоднородные узлы.

OPC-стандарт можно рассматривать как "универсальный переходник". Если мы заменяем какой-нибудь компонент клиентского приложения, то нет необходимости корректировать серверное приложение, поскольку COM обеспечивает эффективное управление изменением программы. Если мы хотим добавить в систему новое оборудование, то достаточно его включить в OPC-сервер. При этом клиентские программы не требуют изменений, в т. ч. для SCADA-систем.

В настоящее время, консорциум OPC Foundation набирает силу в разработке открытых промышленных стандартов на основе OPC-стандарта на базе PC и операционных систем Microsoft. Как указывалось ранее, в состав OPC Foundation входят более 300 членов, среди которых практически все мировые ведущие производители технологического оборудования, систем автоматизированного управления и программного обеспечения. Членами организации являются, например, фирмы Iconics Inc. (США), Wonderware (США), Citect (Австралия), AdAstra (Россия), Siemens (Германия), Rockwell Software (США), Intellution (США), Indusoft Russia (Россия), Fastwel Inc. (Россия), ABB Automation (США), Fieldbus Foundation (США), Toshiba Corp. (Япония), Hitachi (Япония), National Instruments (США) и др. Организация пытается охватить все аспекты, связанные с взаимодействием с технологическим оборудованием. Ведущие производители стараются максимально учесть свой опыт и предоставить абсолютно всё необходимое тому, кто будет использовать OPC. Этот факт показывает большой авторитет OPC-технологии. Это перспективная технология для использования в развитии автоматизированных систем управления.

2.4. Заключение

OPC-технология содержит стандарты, обеспечивающие взаимодействие программных средств промышленной автоматизации. В технологию заложены богатые возможности, которые дают руководителям предприятия возможность интеграции разнородных систем и обеспечивают разработчикам свободу выбора с применением OPC-спецификаций, позволяют не задумываться по поводу поддержки аппаратуры завтрашнего дня. Консорциум OPC Foundation создал интерфейс, который объединяет все компоненты систем автоматизации и визуализации ведущих производителей. OPC станет технологией, которую не сможет проигнорировать ни один производитель средств промышленной автоматизации.

Глава 3. Краткий обзор SCADA-системы GeniDAQ

Рассмотрим основные отличительные особенности SCADA-системы GeniDAQ и ее место на рынке SCADA-систем.

3.1. Почему мы рассматриваем GeniDAQ?

Производитель SCADA-системы GeniDAQ — американское отделение фирмы Advantech, которая является известным производителем компьютеров и электроники для промышленной автоматизации. Отличительной чертой SCADA-системы GeniDAQ является то, что Вы можете получить полноценную SCADA-систему (Runtime-версию) для Windows всего за несколько сотен долларов. Даже "доморощенные" пакеты с неясным будущим за такую цену уже давно никто не предлагает. А почувствовать себя честным пользователем лицензионно чистой копии добротного сделанного продукта фирмы с мировым именем почти даром — просто приятно. Секрет низкой цены в этом случае раскрывается просто — SCADA-система GeniDAQ предназначена для программной поддержки аппаратуры фирмы Advantech и, в первую очередь, содержит драйверы именно для нее. Вместе с тем, никто не запрещает использовать SCADA-систему GeniDAQ и с оборудованием других изготовителей. Последнее объясняется тем, что SCADA-пакет GeniDAQ полнофункционально поддерживает OPC-спецификацию. С этой точки зрения, SCADA-систему GeniDAQ можно рассматривать в качестве OPC-клиента. Большинство ведущих производителей оборудования в мире поставляет на рынок свои аппаратные средства вместе с соответствующими OPC-серверами. С помощью OPC-технологии можно легко создавать приложения, которые работают не только с устройствами фирмы Advantech, но и с аппаратурой любых других фирм.

Другой важной отличительной чертой SCADA-пакета GeniDAQ является прекрасно продуманный интерфейс пользователя. Намеренно сократив число "степеней свободы" в инструментальной части пакета и написав прекрасную справку (Help, помощь), авторы пакета создали уникальный по простоте освоения программный продукт. В худшем случае, уже через несколько часов знакомства с пакетом, Вы сможете написать что-нибудь содержательное и работающее. Если же это окажется не так, то Вам стоит задуматься о смене профессиональной ориентации. Прекрасно выполненная демоверсия пакета одновременно служит и хорошим учебным пособием — разобравшись в работе десятка примеров несложных программ, можно спокойно принять решение о приобретении полной версии [11].

3.2. Системная архитектура GeniDAQ

SCADA-пакет GeniDAQ имеет открытую архитектуру[5, 12], что обеспечивает наибольшую гибкость и эффективность (рис. 3.1).

Центр обработки данных является основным информационным хранилищем в GeniDAQ и представляет собой набор библиотек динамической компоновки (DLL), которые предназначены для размещения и хранения всех данных, связанных с работой приложения под управлением исполнительного окружения GeniDAQ. Результаты выполнения всех функциональных блоков и данные, вводимые пользователем с помощью элементов управления в экранных формах, передаются в центр обработки данных.

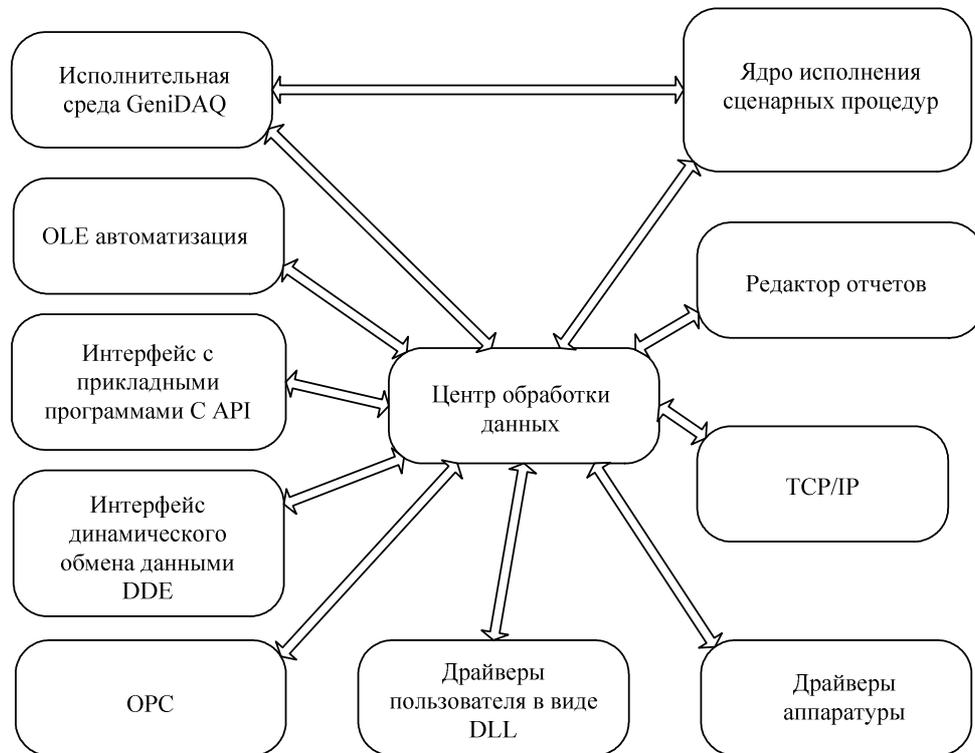


Рис. 3.1. Системная архитектура SCADA-пакета GeniDAQ

Центр обработки данных располагается в физической (не виртуальной) оперативной памяти для обеспечения наиболее быстрого сохранения и извлечения данных. Ключом для поиска информационного объекта в центре обработки данных является имя тега. Для увеличения скорости доступа к информации в центре обработки данных могут быть использованы эффективные алгоритмы, реализация которых имеется в библиотеке классов Microsoft Foundation Classes (MFC). Каждый объект в центре обработки данных имеет имя тега, идентификатор функционального блока, период обновления информации, счетчик доступа и значение, связанное с объектом. Другие приложения Windows могут использовать имя тега для получения идентификатора функционального блока из центра обработки данных. Идентификатор функционального блока, наряду с именем тега, также может

применяться в качестве ключа для поиска объекта в центре обработки данных. Центр обработки данных поддерживает ряд способов взаимодействия пакета с другими приложениями Windows:

- интерфейс прикладного программирования C API;
- интерфейс динамического обмена данными DDE;
- интерфейс связывания и внедрения (встраивания) объектов OLE Automation;
- OPC-интерфейс для подключения к OPC-серверам;
- интерфейс TCP/IP для организации обмена данными в сети.

Интерфейс прикладного программирования C API представляет собой наиболее эффективное средство взаимодействия между SCADA-пакетом GeniDAQ и другими приложениями, поскольку с его помощью осуществляется взаимодействие между программными компонентами GeniDAQ.

Интерфейс связывания и внедрения объектов OLE Automation предназначен для организации взаимодействия между GeniDAQ и другими приложениями Windows, поддерживающими механизм OLE. Интерфейс OLE Automation является основой *интерфейса OPC*, который можно рассматривать как дальнейшее развитие OLE Automation. OPC-интерфейс используется для обмена данными между GeniDAQ и OPC-серверами для управления и сбора данных в системах промышленной автоматизации.

С помощью *интерфейса TCP/IP* данные о технологическом процессе можно наблюдать и регистрировать в режиме реального времени в любой точке локальной сети.

Исполнительная среда GeniDAQ обеспечивает возможность переключения нескольких задач в процессе исполнения, а также позволяет осуществлять однократный вызов задач. Этим объясняется способность GeniDAQ одновременно исполнять несколько задач. Поскольку один вызов задачи является наименьшей единицей исполнения, пользователи получают возможность разрабатывать приложение таким образом, чтобы логически и функционально связанные блоки группировались в одной и той же задаче. Использование основного сценария (подробнее см. далее) для управления исполнением задач позволяет вызывать те или иные задачи только при выполнении определенных условий. В результате появляется возможность разработки сложных систем путем объединения простых задач. Простое приложение, содержащее одну задачу, может исполняться без использования сценарных процедур. Имеется возможность определения приоритетов выполнения задач для оптимизации работы системы.

Ядро подсистемы исполнения сценарных процедур пакета GeniDAQ представляет собой набор библиотек динамической компоновки, с помощью которых выполняется предварительная компиляция сценарных процедур на этапе разработки, и исполнение сценариев в процессе выполнения приложения. Бейсик-сценарий обеспечивает возможность не только манипуляции задачами, входящими в стратегию (приложение), но и взаимодействия с Windows-приложениями и другими приложениями посредством механизмов DDE, OLE, OPC и ODBC (SQL). Синтаксис

Бейсик-сценария совместим с Microsoft VBA (Visual Basic for Application, используемым в Excel, Word, Access и т. п.) и Microsoft Visual Basic. Бейсик-сценарий и VBA имеют ряд несовместимых функций, но более 95% функций и процедур абсолютно идентичны. При использовании в Бейсик-сценарии базовых функций имеется возможность компиляции и исполнения программ, написанных на Visual Basic, без каких-либо изменений. Номера ошибок и сообщения об ошибках в Бейсик-сценарии также совместимы с имеющимися в Visual Basic. В состав Бейсик-сценария входит редактор, позволяющий разрабатывать собственные средства взаимодействия с оператором. Средство программирования сценарных процедур делает пакет GeniDAQ одним из наиболее удобных и современных инструментов для разработки программного обеспечения верхнего уровня систем сбора данных и оперативного диспетчерского управления (SCADA).

Драйверы в виде библиотек динамической компоновки. В пакете GeniDAQ, как и в более ранних версиях пакета, используются загружаемые драйверы в виде библиотек динамической компоновки. Для обмена информацией с исполнительной средой GeniDAQ в драйверах используется интерфейс прикладного программирования С API. *Драйверы устройств* могут непосредственно обмениваться данными с центром обработки данных посредством вызовов функций языка С. Таким же способом может осуществляться информационный обмен между GeniDAQ и другими приложениями Windows.

Редактор отчетов является отдельным приложением, которое полностью независимо от исполнительной среды. Редактор отчетов обеспечивает пользователю возможность формирования отчетов и вывода их на экран монитора или печатающее устройство.

Глава 4. Учебник по SCADA-системе GeniDAQ

Данная глава является основной главой первой части учебного пособия. В ней с использованием демонстрационных примеров приводятся общие правила выполнения основных операций редактора задач, редактора форм отображения и редактора сценариев SCADA-пакета GeniDAQ [13].

Совет

На прилагаемом DVD-диске содержатся файлы для инсталляции демонстрационной версии SCADA-системы GeniDAQ и указания по ее инсталляции. Настоятельно советуем вам установить SCADA-систему и пользоваться ею при изучении материала — это важно.

Для создания приложений (стратегий) и окон отображения с помощью редактора задач и редактора форм отображения следует использовать мышь, что позволит значительно повысить степень интуитивного восприятия предоставляемого GeniDAQ графического интерфейса. Далее будет использоваться следующая терминология, относящаяся к способам работы с указательными устройствами в среде операционной системы Windows.

Указание объекта заключается в помещении подвижного (не текстового) курсора мыши в область отображения объекта (пиктограммы, кнопки, поля, имени файла и т. п.) на экране монитора путем перемещения мыши.

Щелчок и двойной щелчок — помещение подвижного курсора мыши в область отображения объекта на экране монитора путем перемещения мыши с последующим нажатием и быстрым отпусканием ее левой клавиши (щелчок) или с последовательным двойным нажатием и отпусканием левой клавиши мыши (двойной щелчок). Указанные действия обычно ассоциируются с выбором объекта и с выполнением операции, связанной с данным объектом. Выбор блока в окне редактора задач или элемента отображения в окне редактора форм отображения приведет к выделению выбранного объекта вдоль периметра рамкой. Объект остается выбранным (выделенным) до выполнения следующей операции.

Перенос курсора — перемещение подвижного курсора мыши при одновременном нажатии и удержании ее левой клавиши с последующим отпусканием. При выполнении операции переноса с одновременным выбором объекта происходит его перемещение до отпускания левой клавиши мыши. Операция переноса курсора может быть использована также для выделения нескольких объектов в окне редактора задач или редактора форм отображения.

Использование справочной системы. В процессе работы с GeniDAQ имеется возможность получения справочной информации путем выполнения команды **Help | Help Topics** строителя стратегии. Кроме того, в каждом окне диалога имеется кнопка **Help**, позволяющая вызвать относящуюся к текущему диалогу

справочную информацию. Контекстная справка, если она предусмотрена, может быть вызвана путем выбора кнопки на панели инструментов с изображением стрелки и вопросительного знака.

Учебные стратегии GeniDAQ. Для быстрого освоения приемов работы со SCADA-системой GeniDAQ в комплект поставки включены девять учебных приложений (стратегий), которые после инсталляции SCADA-системы помещаются в папку \Program Files\Advantech\GeniDAQ\Strategy. В учебных стратегиях используется программный эмулятор сигналов **Advantech DEMO I/O=1H**, который позволяет создавать и запускать на исполнение демонстрационные и обучающие приложения без применения каких-либо дополнительных аппаратных средств. В этой же папке имеется и еще ряд стратегий, изучение которых представляет интерес. Далее, с целью обучения, приводится описание всех учебных приложений и, дополнительно, разработанных автором некоторых других стратегий (приложений). Такой подход обеспечивает быстрое формирование у пользователя начальных навыков работы со SCADA-системой.

4.1. Занятие 1 "Демонстрация базовых приемов работы — одна задача реального времени с отображением результатов ее работы в одном экранном окне"

Цель занятия состоит в демонстрации базовых приемов работы со SCADA-системой GeniDAQ. Далее показывается простой способ использования средств, предоставляемых GeniDAQ, для сбора в реальном масштабе времени измерительной информации и отображения ее в экранном окне на графике зависимости от времени и на цифровом индикаторе. Для этого используются программный эмулятор сигналов **Advantech I/O**, блок аналогового ввода для сбора измерительной информации и два индикатора — график зависимости от времени и цифровой индикатор.

4.1.1. Используемый инструментарий

Используемый инструментарий. Для построения пользовательского приложения, выполняющего требуемые функции, используется построитель приложений GeniDAQ Builder и программный эмулятор **Advantech DEMO I/O**. Процесс построения приложения состоит из двух этапов:

- собственно проектирования приложения (в SCADA-системе GeniDAQ этот этап называют построением *стратегии*);
- разработка визуального отображения работы созданного приложения для оперативного персонала (в SCADA-системе GeniDAQ этот этап называют построением *форм отображения*).

На первом этапе построитель приложений использует *редактор задач*, имеющий собственное окно и панель инструментов (рис. 4.1).

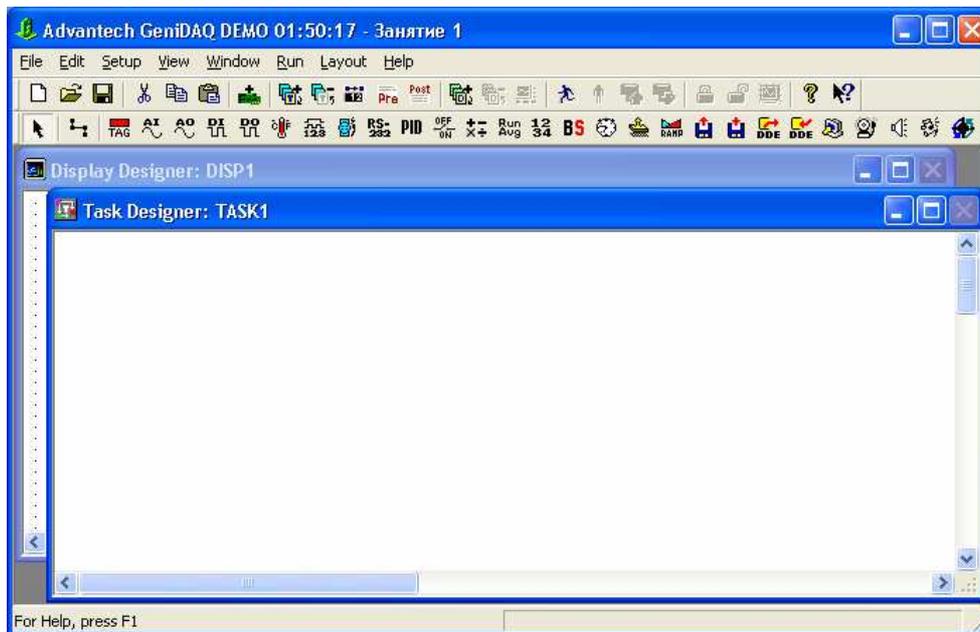


Рис. 4.1. Главное окно построителя приложений с окном **Task Designer: TASK1** и панелью инструментов редактора задач

На втором этапе построитель приложений использует *редактор форм отображения*, который также использует собственное окно и панель инструментов (рис. 4.2).

Редактор задач использует визуальную (информационно-поточную) модель программирования, которая значительно удобнее, чем традиционная линейная архитектура текстовых языков программирования. При проектировании приложения для сбора данных и управления пользователь (разработчик) создает блок-схему стратегии. При этом он не должен уделять особого внимания различным логическим и синтаксическим соглашениям, принятым в стандартных языках программирования. При проектировании приложения разработчик просто выбирает объекты (пиктограммы функциональных блоков на панели инструментов редактора задач) и соединяет их проводниками для передачи данных от одного блока к другому. Каждый функциональный блок, представлен пиктограммой на панели инструментов редактора задач и предназначен для выполнения соответствующей встроенной функции обработки данных, поступающих от аппаратуры или вводимых пользователем. Ряд блоков позволяет организовывать взаимодействие непосредственно с низкоуровневыми драйверами аппаратуры.

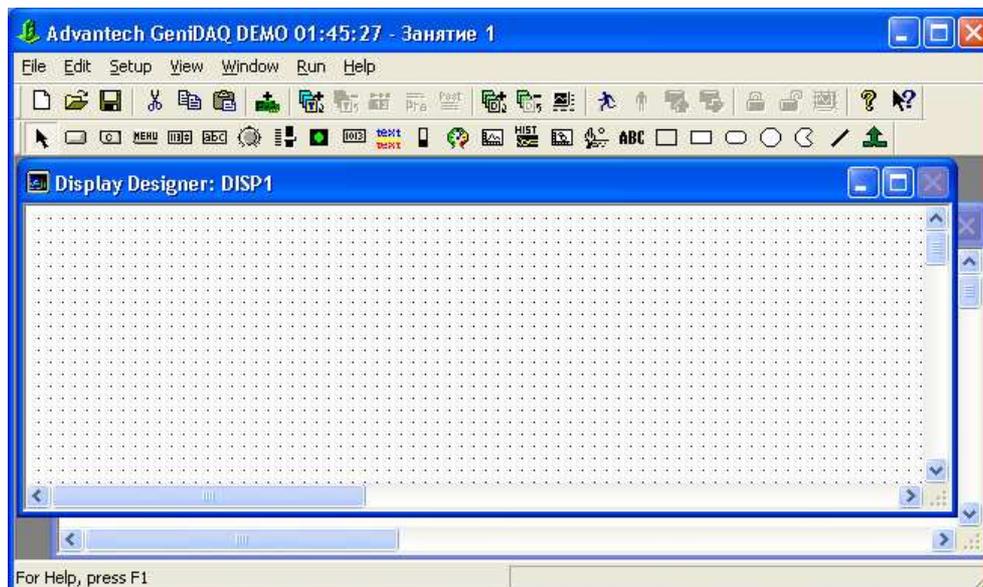


Рис. 4.2. Главное окно построителя приложений с окном **Display Designer: DISP1** и панелью инструментов редактора форм отображения

Ранее уже упоминалось, что редактор задач состоит из двух основных компонент: панели инструментов и окна. Для создания стратегии в клиентскую область окна редактора задач с помощью операции "перенести и оставить" (Drag and Drop) помещаются требуемые функциональные блоки. При этом размещение функционального блока выполняется путем щелчка левой кнопкой мыши по его пиктограмме на панели инструментов, последующем перемещении курсора мыши в требуемую позицию окна редактора задач и повторного щелчка левой кнопкой мыши. После размещения в окне всех необходимых функциональных блоков и их конфигурирования устанавливаются связи между ними в соответствии с требованиями реализуемого алгоритма сбора данных и управления. Для этой цели используется проводник, представленный соответствующей пиктограммой на панели инструментов редактора задач.

В рамках данного занятия используется один функциональный блок редактора задач — *блок аналогового ввода*. Данный функциональный блок предназначен для приема информации от устройств, имеющих подсистему ввода аналоговых сигналов, и передачи указанных сигналов другим функциональным блокам и элементам отображения. Двойной щелчок левой клавишей мыши на пиктограмме блока аналогового ввода в окне редактора задач приводит к появлению диалогового окна настройки параметров блока. Пример диалогового окна приведен на рис. 4.3. Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров блока аналогового ввода будут также рассмотрены в последующих примерах.

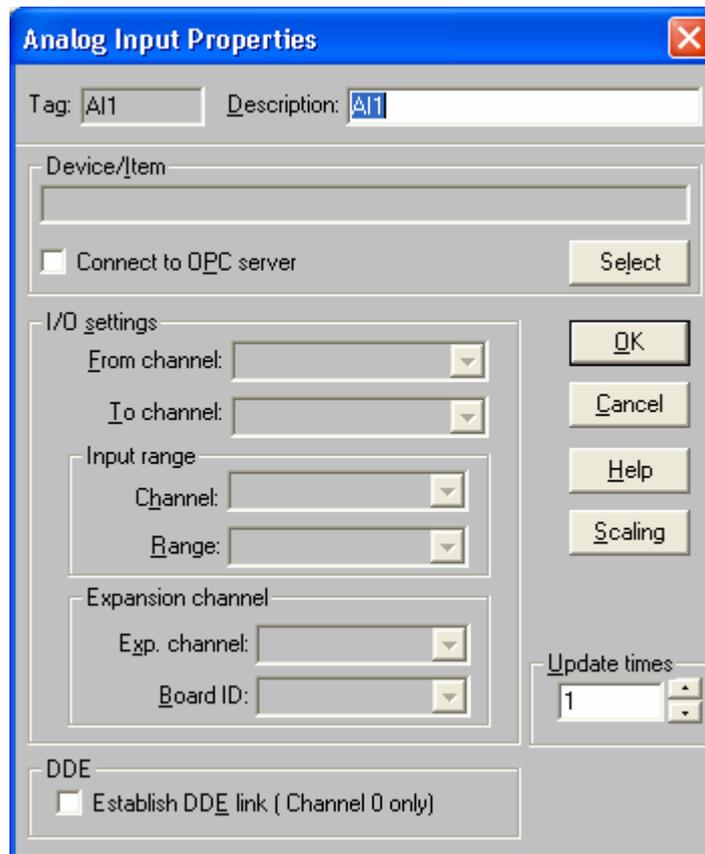


Рис. 4.3. Вид диалогового окна настройки блока аналогового ввода

Редактор форм отображения предназначен для создания графических мнемосхем автоматизированных рабочих мест оператора, обеспечивающих динамическое представление информации о контролируемом технологическом процессе в удобной для восприятия форме. Кроме того, редактор обеспечивает возможность использования растровых изображений, создаваемых разработчиком, в качестве фоновых рисунков. Оператор может не только следить за ходом контролируемого процесса, но и осуществлять управление его параметрами в процессе исполнения стратегии, спроектированной с помощью редактора задач.

По завершении или во время разработки стратегии может быть принято решение о том, какая информация и в каком виде должна быть представлена оператору в окне редактора форм отображения в процессе исполнения стратегии. Перед началом создания мнемосхемы рекомендуется уделить серьезное внимание предварительной проработке ее внешнего вида — перечню элементов отображения/управления (динамических элементов), виду фоновых рисунков (статическая часть мнемосхемы),

а также тому, какие именно данные должны быть отображены с помощью динамических элементов.

Редактор форм отображения, как и редактор задач, состоит из двух основных компонент: панели инструментов с пиктограммами динамических элементов (элементов отображения/управления) и окна. Для создания мнемосхемы в клиентскую область окна редактора форм отображения с помощью операции "перенести и оставить" (Drag and Drop) помещаются требуемые динамические элементы. После размещения в окне всех необходимых динамических элементов, их конфигурирования и создания фоновых рисунков создание мнемосхемы закончено и ею можно пользоваться.

В рамках данного занятия используется два динамических элемента редактора форм отображения — цифровой индикатор (Numeric String) и временной график (Realtime Trend Graph).

Цифровой индикатор предназначен для отображения значения параметра на выходе присоединенного функционального блока стратегии, а также для вывода на экран текстовых строк, поступающих с выхода блока процедуры пользователя или блока Бейсик-сценария в процессе исполнения стратегии. Цифровой индикатор может быть помещен в окно формы отображения и связан с выходной переменной функционального блока задачи, входящей в стратегию. Имеется возможность установки требуемых размеров индикатора. Цифровой индикатор может использоваться для отображения вещественных (с плавающей точкой) и целочисленных переменных, а также текстовых строк. Кроме того, имеется возможность установки требуемого количества цифр после десятичной точки при отображении вещественных значений, выбора типа шрифта, его цвета и размера, а также метода выравнивания отображаемого значения или строки. Двойной щелчок левой клавишей мыши на пиктограмме цифрового индикатора в окне редактора форм отображения приводит к появлению диалогового окна настройки параметров индикатора. Пример диалогового окна приведен на рис. 4.4. Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров цифрового индикатора будут также рассмотрены в примерах, следующих далее.

Временной график предназначен для отображения на мнемосхеме (в окне формы отображения) временной зависимости одной или нескольких переменных стратегии. Количество зависимостей величин на выходах функциональных блоков от времени, отображаемых с помощью динамического элемента, может быть неограниченным, однако из практических соображений не рекомендуется строить более восьми временных зависимостей на одном графике. Имеется возможность выбора цвета графика и диапазонов по осям абсцисс (время) и ординат (параметр). Двойной щелчок левой клавишей мыши на пиктограмме временного графика в окне редактора форм отображения приводит к появлению диалогового окна настройки параметров графика. Пример диалогового окна приведен на рис. 4.5. Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров временного графика будут рассмотрены в примерах, следующих далее.

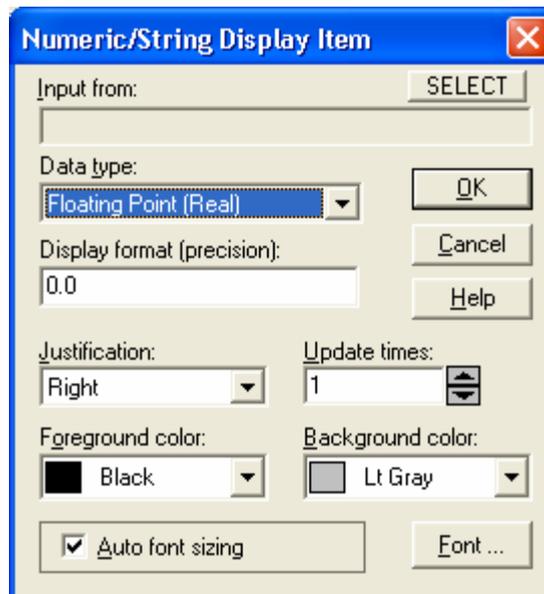


Рис. 4.4. Вид диалогового окна для задания параметров цифрового индикатора

Программный эмулятор *Advantech DEMO I/O* представляет собой программный эмулятор платы ввода/вывода, который позволяет проверить большинство функций без использования каких-либо аппаратных средств ввода-вывода. Эмулятор имеет в своем составе четыре канала аналогового ввода и два канала цифрового ввода — вывода. Канал 0 аналогового ввода моделирует синусоидальный сигнал ± 5 вольт, канал 1 — сигнал прямоугольной формы ± 5 вольт, канал 2 — пилообразный сигнал ± 5 вольт и канал 3 — случайный сигнал. Каждый используемый эмулятор должен иметь собственный уникальный базовый адрес. Устанавливайте базовый адрес для каждого эмулятора ввода/вывода в соответствующем диалоговом окне.

4.1.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. *Первое действие* — запустите построитель приложений и создайте новое приложение. После включения компьютера на экране компьютера появляется запрос на ввод атрибутов пользователя. Запустите построитель приложений двойным щелчком левой кнопкой мыши на ярлыке **GeniDAQ Builder 4.25**, расположенном на рабочем столе. В результате на экран будет выведено главное окно построителя приложений (рис. 4.6). Для запуска построителя задач в демонстрационном режиме последовательно нажмите кнопки **Exit** и **OK**. Демонстрационный режим отличается непрерывным двухчасовым интервалом работы и меньшим количеством входных и выходных каналов. В нашем случае

это несущественно. Нажмите кнопку **New** на панели инструментов (ее легко найти, так как все кнопки снабжены всплывающими подсказками). После этого в главном окне построителя приложений появится диалоговое окно настройки, которое может быть настроено так, как это показано на рис. 4.7. При работе в учебной лаборатории в поле **Location** качестве каталога следует использовать каталог, к которому имеется полный доступ. После нажатия кнопки **OK** и подтверждения необходимости создания каталога **Занятие 1**, который до этого не существовал, в главном окне построителя задач появятся пустые окна редактора задач **TASK1** и редактора форм отображения **DISP1**. Произведите щелчок мышью в области окна редактора задач и окно с заголовком **Task Designer: TASK1** будет выдвинуто на передний план (см. приведенный ранее рис.4.1). В окне построителя задач появится еще одна панель инструментов, содержащая пиктограммы встроенных функциональных блоков редактора задач.

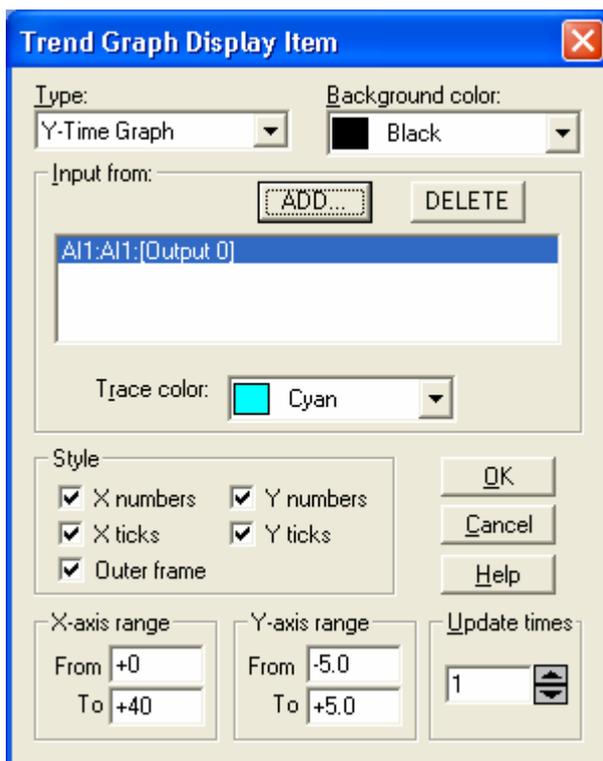


Рис. 4.5. Пример диалогового окна настройки графика времени

Примечание

Атрибуты пользователя следует получить у преподавателя. Пользователь должен обладать правами администратора.

2. *Второе действие* — задайте временные параметры запуска приложения. Для этого нажмите кнопку **Task Properties** на панели инструментов или выполните команду **Setup | Task Properties**, в появившемся окне задайте требуемое значение периода (рис. 4.8) и нажмите кнопку **ОК**. В нашем примере задан период запуска приложения 100 миллисекунд.
3. *Третье действие* — разместите в окне редактора задач блок аналогового ввода и настройте его. На панели инструментов выберите кнопку с мнемоникой **AI**, поместите на нее курсор мыши, щелкните левой кнопкой мыши, переместите мышь в окно редактора задач и еще раз щелкните левой кнопкой мыши. При этом в окне редактора задач появится блок аналогового ввода с позиционным обозначением **AI1**. Данную операцию принято называть "перенести и оставить" (Drag and Drop) — рис. 4.9. Произведите двойной щелчок левой кнопкой мыши над блоком аналогового ввода и появится диалоговое окно его настройки. Для выбора источника аналогового сигнала, если для блока аналогового ввода не указано связанное с ним устройство (поле **Device/Item** пусто), нажмите кнопку **Select**, в появившемся окне диалоговом окне выберите программный эмулятор сигнала (рис. 4.10) и нажмите кнопку **ОК**. В данном случае **1H** является 16-ричным адресом экземпляра эмулятора. В результате диалоговое окно настройки блока аналогового ввода приобретает вид, показанный на рис. 4.11. Нулевые значения каналов в полях **From channel** и **To channel** означают, что в эмуляторе будет использован синусоидальный сигнал с амплитудой 5 вольт и нулевым смещением. Значение в поле **Update times** является делителем, который позволяет вызывать блок аналогового ввода и сканировать соответствующие ему каналы устройства аналогового ввода реже, чем вызывается вся задача, в которую входит данный функциональный блок. Например, пусть задача вызывается (сканируется) один раз в 100 миллисекунд. Для того чтобы блок аналогового ввода, входящий в данную задачу, вызывался один раз в 200 миллисекунд, следует установить в поле **Update times** значение 2. В этом случае значение на выходе блока аналогового ввода будет обновляться через каждые два вызова задачи, содержащей блок. Делитель может изменяться в диапазоне от 1 до 32767. В нашем примере задача и блок аналогового ввода выполняются синхронно, с одинаковым периодом 100 миллисекунд. Для сохранения значений параметров блока аналогового ввода нажмите кнопку **ОК**.

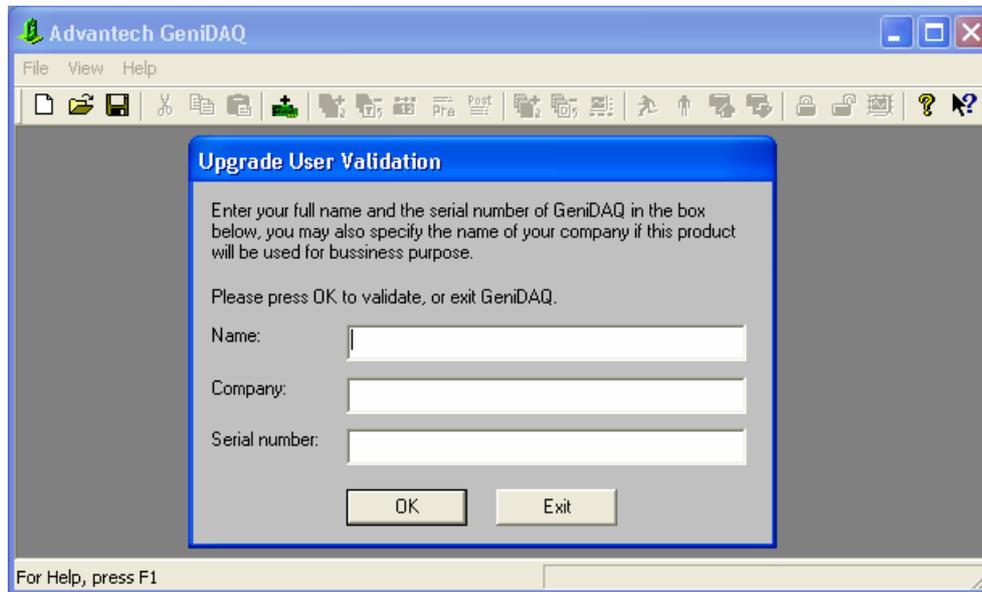


Рис. 4.6. Главное окно построителя приложений

4. *Четвертое действие* — перейдите в окно редактора форм отображения и сконфигурируйте его. Чтобы сделать активным окно **DISP1** редактора форм отображения щелкните по этому окну левой кнопкой мыши. При этом в верхней части главного окна появится панель инструментов редактора форм отображения (см. приведенный ранее рис. 4.2). На панели инструментов выберите кнопку **RealTime Trend Graph** (временной график) и с помощью мыши перетащите ее и положите во внутреннюю область окна редактора форм отображения (рис. 4.12). Для настройки свойств временного графика щелкните дважды левой кнопкой мыши над пиктограммой временного графика. В результате появится диалоговое окно настройки, показанное ранее на рис. 4.5. Для привязки графика к источнику информации в этом окне нажмите кнопку **ADD** и в появившемся окне выполните необходимую настройку (рис. 4.13). Затем с помощью кнопок **OK** закройте это и предыдущее окна. Аналогичным образом перетащите в окно редактора форм отображения и настройте цифровой индикатор. С этой целью на панели инструментов выберите кнопку **Numeric String** (цифровой индикатор) и с помощью мыши перетащите ее и положите во внутреннюю область окна редактора форм отображения (рис. 4.14). Для настройки свойств цифрового индикатора щелкните дважды левой кнопкой мыши над пиктограммой цифрового индикатора. В результате появится диалоговое окно настройки, показанное ранее на рис. 4.4. Для привязки графика к источнику информации в этом окне нажмите кнопку **SELECT** и в появившемся окне выполните необходимую настройку (рис. 4.15). Затем с помощью кнопок **OK** закройте это и предыдущее окна.

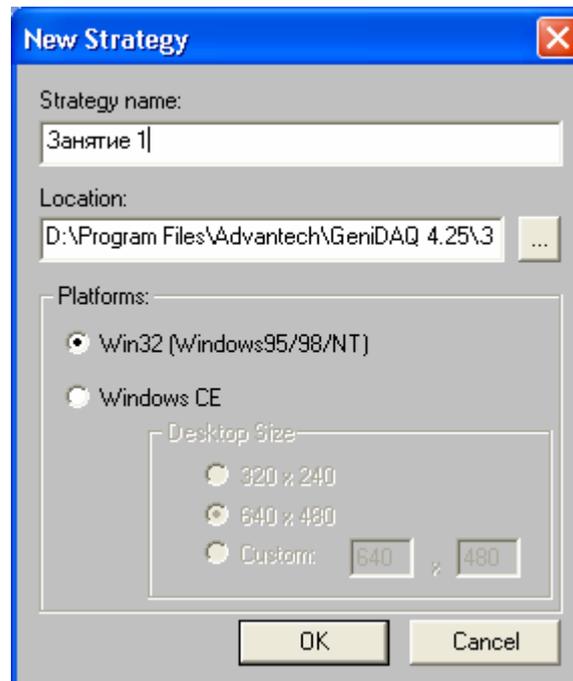


Рис. 4.7. Окно настройки приложения

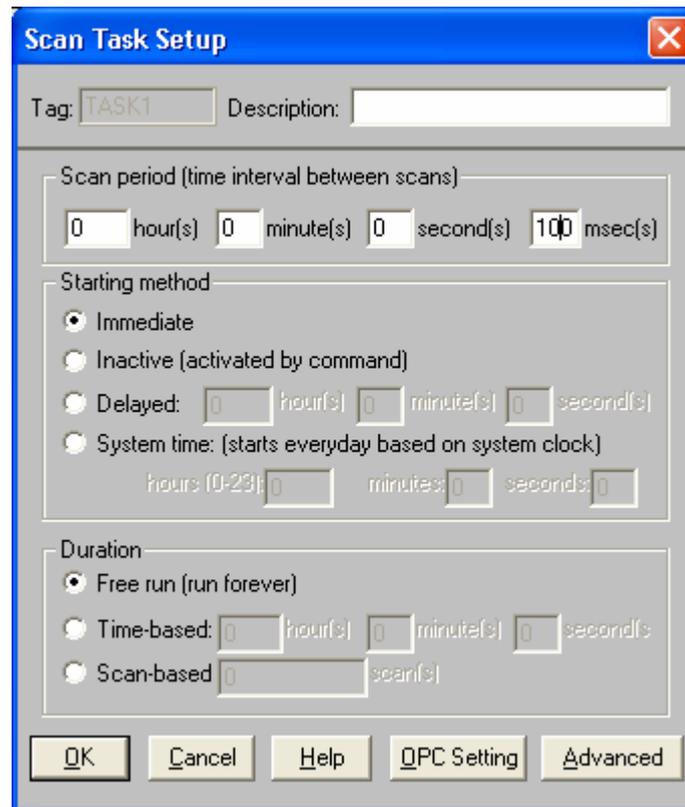


Рис. 4.8. Задание временных параметров запуска приложения

5. *Пятое действие* — сохраните созданный проект (стратегию). Для этого выполните команду **File | Save As**, в появившемся окне диалога укажите необходимую информацию (рис. 4.17) и нажмите кнопку **Сохранить**.
6. *Шестое действие* — запуск и останов проекта. После сохранения созданного проекта имеется возможность немедленного запуска созданного проекта с помощью кнопки **Start** на панели инструментов (рис. 4.17). Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
7. *Седьмое действие* — завершение работы *GeniDAQ*. Завершите работу приложения. Для этого выполните команду **File | Exit**.

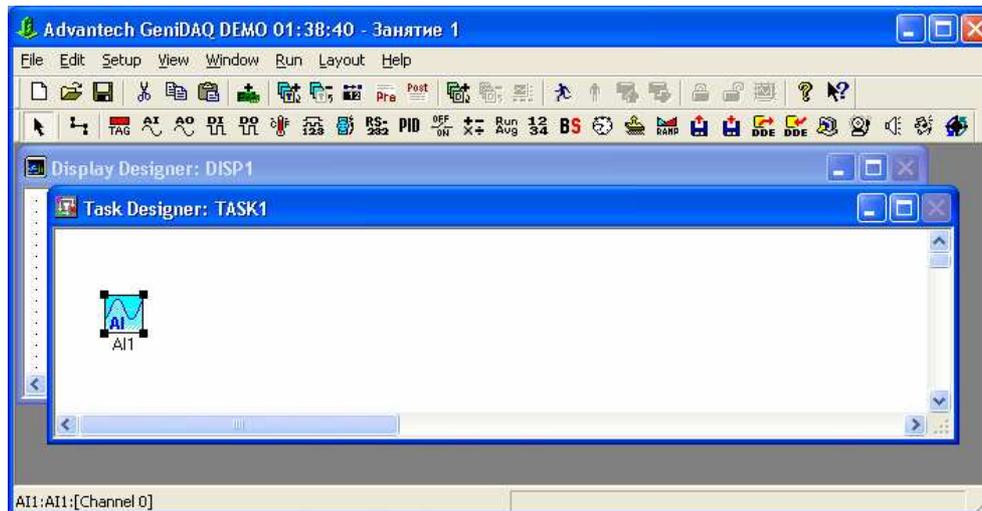


Рис. 4.9. Добавление блока аналогового ввода

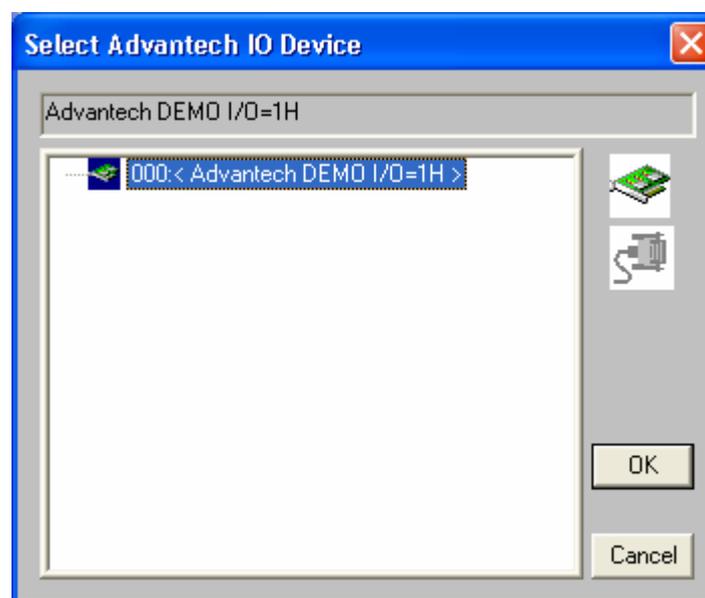


Рис. 4.10. Добавление программного эмулятора ввода/вывода

4.1.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

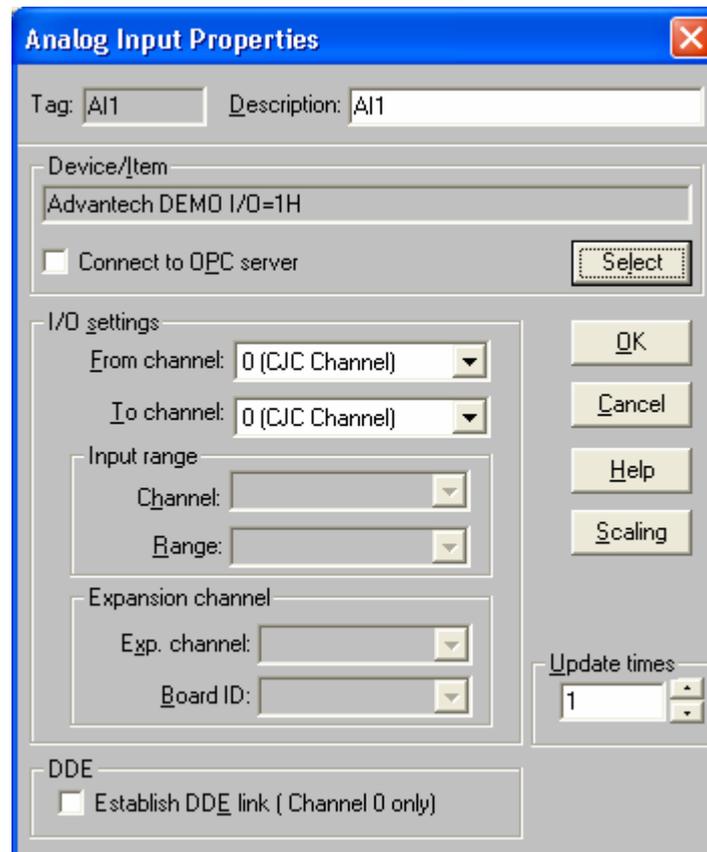


Рис. 4.11. Параметры блока аналогового ввода

Упражнение 4.1. Повтор создания и тестирования приложения "Занятие 1"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 1**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

Упражнение 4.2

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 0.2 секунды. С помощью эмулятора и четырех блоков аналогового ввода, помещенных в окно редактора задач, смоделируйте соответственно гармонический, прямоугольный, пилообразный и случайный сигналы. В окне редактора форм отображения покажите указанные процессы на общем графике времени и с помощью цифровых индикаторов. Снабдите элементы отображения поясняющими надписями.

4.2. Занятие 2 "Переключение окон отображения"

Цель занятия состоит в изучении правил разработки приложения с несколькими окнами отображения и способа переключения окон отображения в процессе выполнения приложения. Для этого используется программный эмулятор сигналов **Advantech IO**, блок аналогового ввода для сбора измерительной информации с канала 1 программного эмулятора и два индикатора для отображения получаемой информации в разных окнах отображения — в виде временного графика и цифрового индикатора. Новым является то, что для переключения окон отображения используется кнопка меню.

4.2.1. Используемый инструментарий

Используемый инструментарий. Новым на данном занятии является использование для построения стратегии *командной кнопки (кнопки меню)* редактора форм отображения. Данный элемент управления предназначен для создания командных кнопок в окнах отображения, позволяющих управлять процессом исполнения стратегии. С помощью кнопки меню можно реализовать одну из трех функций **Function**: *действие Action, переключение окна отображения Display switching* или *управление задачей Task control*.

Функция **Action** позволяет выбрать одну из следующих операций при нажатии на кнопку меню: **NONE** — функция выключена; **START** — запуск стратегии на исполнение; **STOP** — завершение исполнения стратегии; **PAUSE** — приостановка исполнения стратегии; **RESUME** — возобновление исполнения после приостановки; **CLOSE** — завершение исполнения стратегии и завершение сеанса исполнительской среды; **LOCK** — блокировка пунктов меню и других органов управления Windows; **UNLOCK** — разблокировка и др. При выборе второй функции нажатие на кнопку во время исполнения стратегии приведет к выдвиганию на передний план окна отображения, идентификатор которого выбран в поле **Display Switch** диалоговой панели настройки параметров элемента управления. Использование именно функции **Display switching** и демонстрируется на данном занятии. Двойной щелчок левой клавишей мыши на пиктограмме кнопки меню в окне редактора форм отображения приведет к появлению диалоговой окна настройки параметров кнопки. Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров кнопки меню будут рассмотрены в нижеследующих примерах.

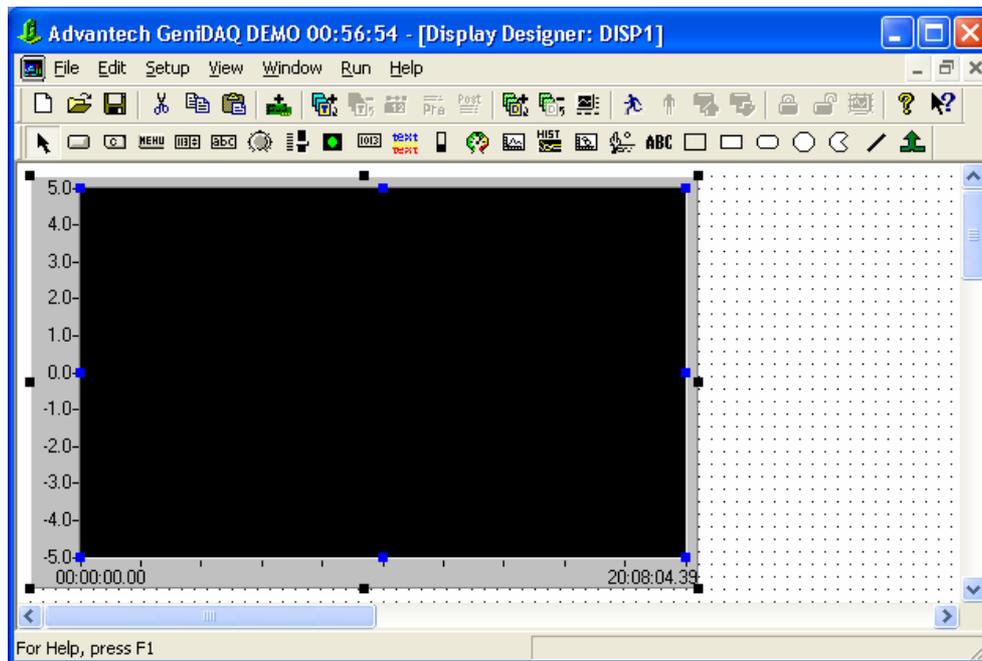


Рис. 4.12. Элемент отображения временного графика

4.2.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.18.
2. Выполните *второе действие* из занятия 1. В нашем примере также задан период запуска приложения 100 миллисекунд.
3. Выполните *третье действие* из занятия 1. Сконфигурируйте блок аналогового ввода в соответствии с рис. 4.19. Единичные значения номеров каналов **From channel** и **To channel** означают, что в эмуляторе будет использован меандр (прямоугольный сигнал) со значениями ± 5 вольт. Для сохранения значений параметров блока аналогового ввода нажмите кнопку **OK**.
4. Выполните *четвертое действие* из занятия 1 в части графика времени. Сконфигурируйте график времени в соответствии с рис. 4.20, а затем с помощью кнопок **OK** закройте окна.
5. *Пятое действие.* Выполните команду **File | Add/Delete | Add Display**. В главном окне построителя приложений появится еще одно окно отображения с заголовком **Display Designer: DISP2**. Поместите в это окно цифровой индикатор аналогично

тому, как это указано в действии 4 из занятия 1 и сконфигурируйте цифровой индикатор в соответствии с рис. 4.21.

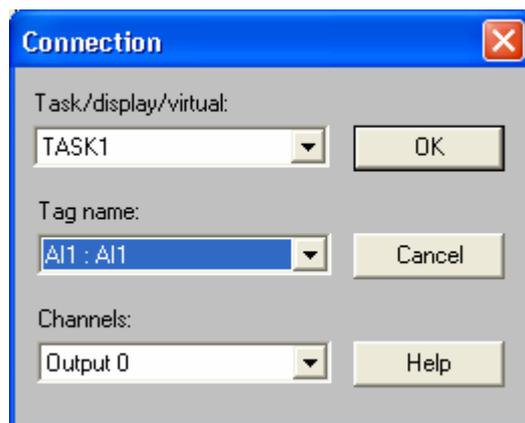


Рис. 4.13. Задание отображаемого параметра для графика времени

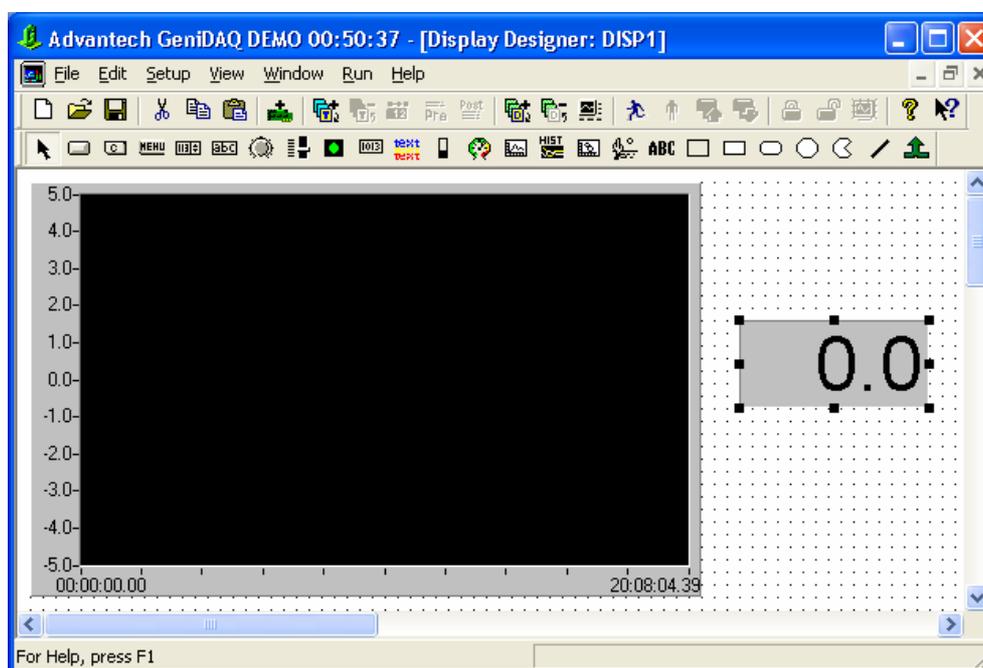


Рис. 4.14. Цифровой индикатор

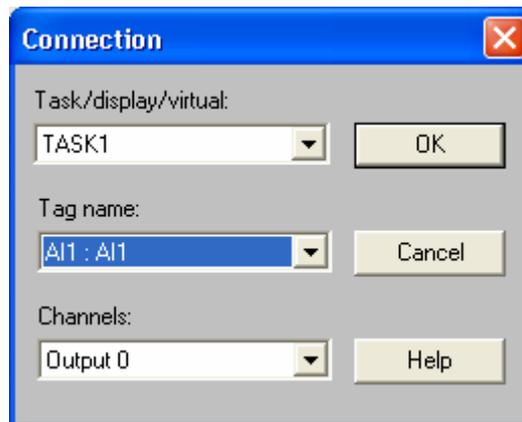


Рис. 4.15. Источник информации для цифрового индикатора

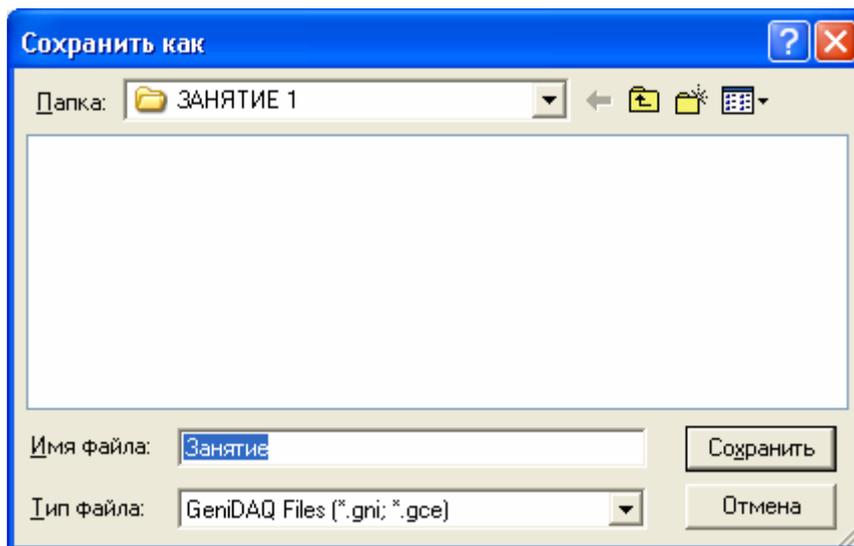


Рис. 4.16. Сохранение созданного программного проекта (стратегии)

6. *Шестое действие.* Перетащите в окно отображения **DISP2** кнопку меню **Menu Button** (рис. 4.22). Сконфигурируйте ее в соответствии с рис. 4.23.
7. *Седьмое действие.* Активизируйте окно отображения **DISP1** и перетащите в него кнопку меню **Menu Button** (рис. 4.24). Сконфигурируйте ее в соответствии с рис. 4.25.
8. *Восьмое действие.* Сохраните созданный проект (стратегию) аналогично пятому действию из занятия 1.

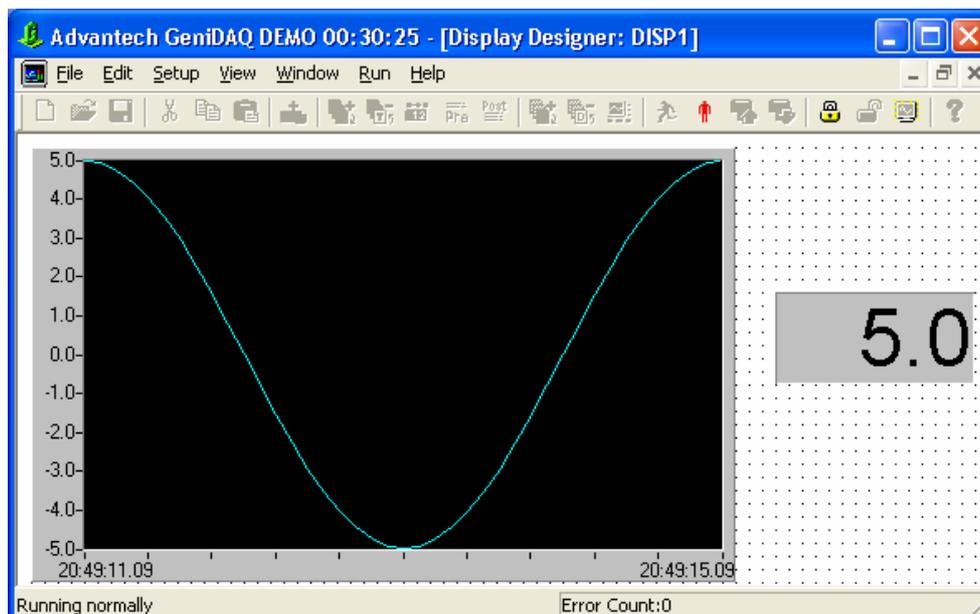


Рис. 4.17. Демонстрация работы проекта

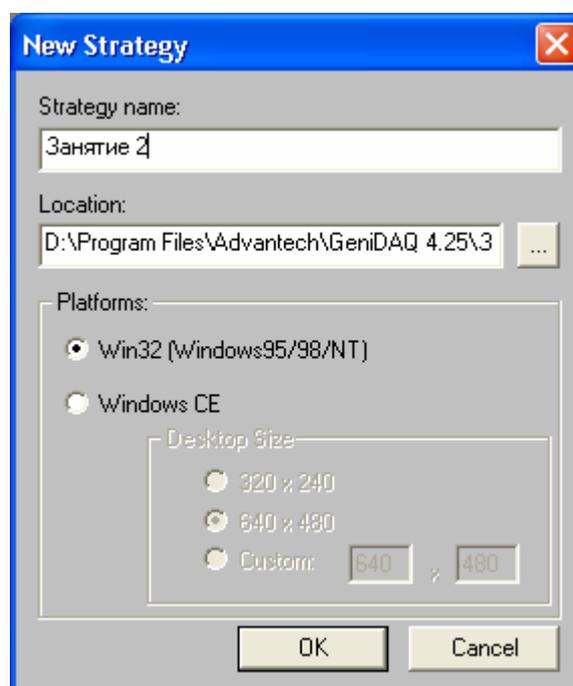


Рис. 4.18. Конфигурирование параметров создаваемого приложения

9. *Девятое действие.* После сохранения файлов созданного проекта имеется возможность немедленного запуска созданного проекта с помощью кнопки **Start** на панели инструментов (рис. 4.26). Для перехода в окно отображения **DISP2** в процессе работы приложения достаточно нажать кнопку меню **B DISP2**. Результат такого действия показан на рис. 4.27. Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
10. *Десятое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.



Рис. 4.19. Конфигурирование блока аналогового ввода

4.2.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

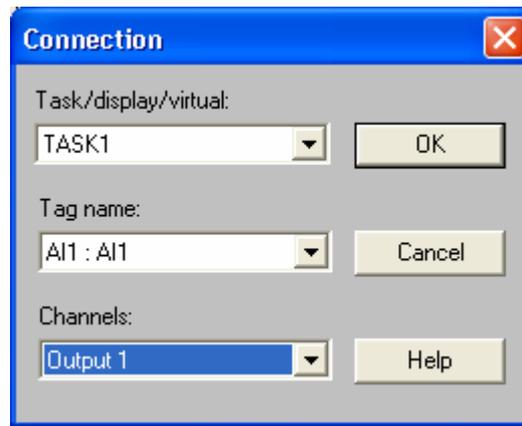


Рис. 4.20. Конфигурирование графика времени

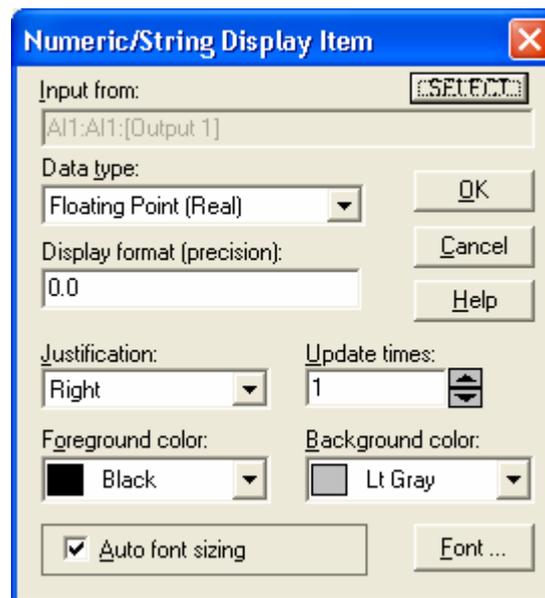


Рис. 4.21. Конфигурирование цифрового индикатора

Упражнение 4.3. Повтор создания и тестирования приложения "Занятие 2"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 2**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

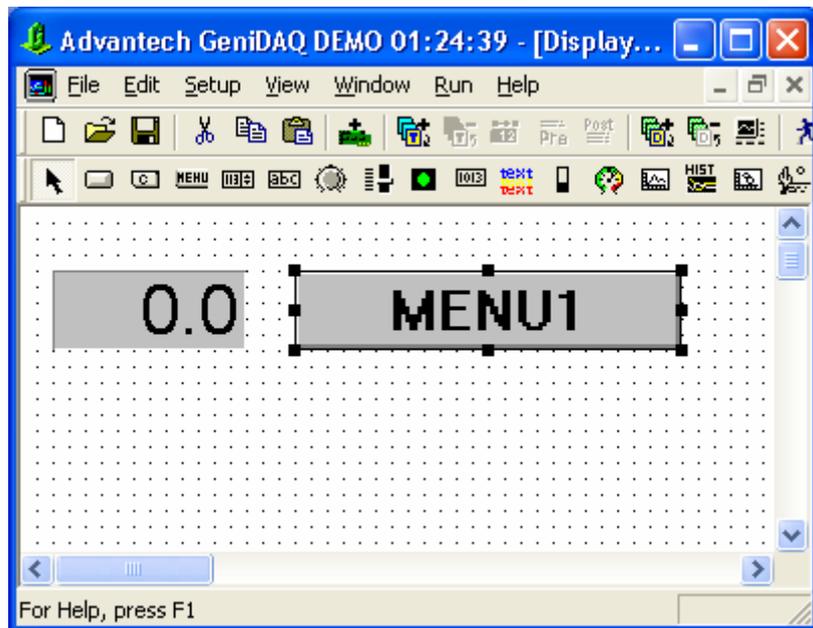


Рис. 4.22. Создание кнопки меню в окне DISP2

Упражнение 4.4

Разработайте стратегию, обладающую следующими возможностями. Создайте три окна редактора задач с периодами запуска 0.5, 1.0 и 1.5 секунды соответственно. С помощью эмулятора и блоков аналогового ввода, помещенных в окна редактора задач, смоделируйте соответственно гармонический, прямоугольный и пилообразный сигналы. Создайте четыре окна отображения. В первом и третьем окнах на графиках времени отобразите выходы блоков аналогового ввода в виде синусоидального и пилообразного сигналов. Во втором окне отобразите с помощью цифрового индикатора выход блока аналогового сигнала в виде меандра. Четвертое (главное) окно отображения должно содержать общий график времени и цифровые индикаторы для отображения выходов блоков аналоговых входов. После запуска стратегии должно отображаться главное окно. Предусмотреть переключение между окнами отображения по кольцу и из каждого окна отображения в главное окно отображения.

Упражнение 4.5

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 0.5 секунды. С помощью эмулятора и четырех блоков аналогового ввода, помещенных в окна редактора задач, смоделируйте соответственно гармонический, прямоугольный, пилообразный и случайный сигналы. Создайте пять окон отображения. В первом окне редактора форм отображения покажите указанные процессы на общем графике времени, а в остальных четырех отобразите их с

помощью цифровых индикаторов. После запуска стратегии должно отображаться окно с временным графиком. Предусмотреть переключение между окнами отображения по кольцу вправо (**DISP1-DISP2-DISP3-DISP4-DISP5-DISP1-...**). Снабдите элементы отображения поясняющими надписями.

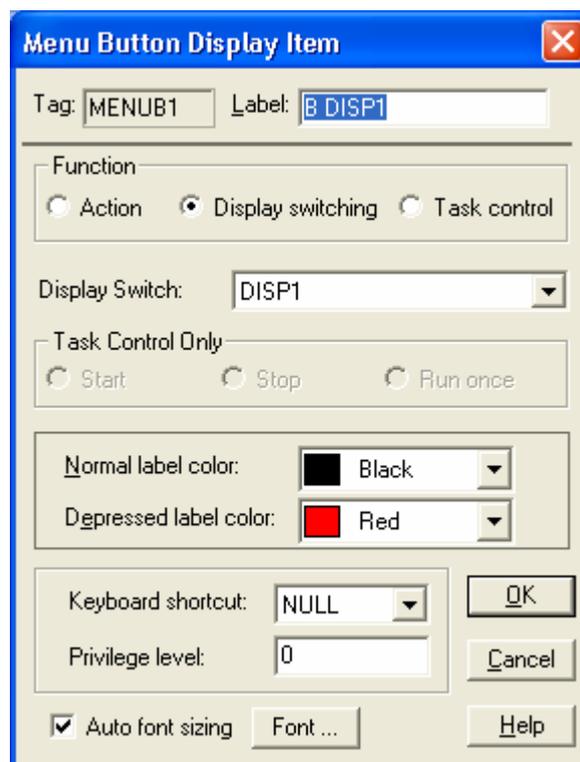


Рис. 4.23. Конфигурирование кнопки меню в окне DISP2

Упражнение 4.6

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 0.1 секунды. С помощью эмулятора и четырех блоков аналогового ввода, помещенных в окно редактора задач, смоделируйте соответственно гармонический, прямоугольный, пилообразный и случайный сигналы. Создайте пять окон отображения. В первом окне редактора форм отображения покажите указанные процессы на общем графике времени, а в остальных четырех отобразите их с помощью цифровых индикаторов. После запуска стратегии должно отображаться окно с временным графиком. Предусмотреть переключение между окнами отображения по кольцу влево (**DISP1-DISP5-DISP4-DISP3-DISP2-DISP1-...**). Снабдите элементы отображения поясняющими надписями.

Упражнение 4.7

Разработайте стратегию, обладающую следующими возможностями. Создайте два окна редактора задач с периодами запуска 1.0 и 1.5 секунды соответственно. С помощью эмулятора и блоков аналогового ввода, помещенных в окна редактора задач, смоделируйте соответственно пилообразный и случайный сигналы. Создайте два окна отображения. В первом окне отобразите пилообразный сигнал с помощью временного графика и цифрового индикатора, а во втором окне аналогично отобразите случайный сигнал. После запуска стратегии должно отображаться окно **DISP1**. Предусмотрите переключение между окнами отображения. Снабдите элементы отображения поясняющими надписями.

4.3. Занятие 3 "Просмотр и изменение порядка выполнения функциональных блоков редактора задач"

Цель занятия состоит в изучении средств просмотра и изменения порядка выполнения функциональных блоков редактора задач. Используются программный эмулятор сигналов **Advantech I/O**, два блока аналогового ввода и два блока вычисления с одним оператором, в одном из которых выполняется сложение выходных сигналов блоков аналогового ввода, а в другом – вычитание. Результаты вычислений представляются в окне отображения с помощью цифровых индикаторов, временного графика и поясняющих строк. Изменяется порядок выполнения функциональных блоков стратегии и оценивается его влияние на результат.

4.3.1. Используемый инструментарий

Используемый инструментарий. Новым на данном занятии является использование для построения стратегии блоков вычисления с одним оператором и средств просмотра и изменения порядка выполнения функциональных блоков стратегии.

Блок вычисления с одним оператором. Данный блок предназначен для выполнения одной математической операции, такой как сложение, вычитание, умножение, деление и т. д. По крайней мере, один функциональный блок стратегии должен быть присоединен к входу блока вычисления с единственным оператором. Значение на выходе присоединенного блока будет являться первым операндом в производимой математической операции. Вторым оператором может быть константа, заданная в соответствующем поле диалоговой панели настройки параметров блока, либо значение на выходе другого присоединенного функционального блока стратегии (рис. 4.28). Результат на выходе блока вычисления с единственным оператором может быть представлен в виде целого (**Integer**) либо вещественного (**Floating Point**) числа. Операторы и функции блока вычисления с единственным оператором приведены в табл. 4.1. Операторы, после которых стоит символ "*", требуют использования в качестве аргументов (операндов) значения целого (**Integer**) типа. Некоторые из перечисленных выше операторов требуют только один аргумент

(операнд). Логические операции (and, or, xor) требуют использования двух аргументов целого типа. Операторы abs, not, inv, sqrt, log, ln, exp, jct требуют использования одного аргумента (целого или с плавающей точкой в зависимости от типа операции).

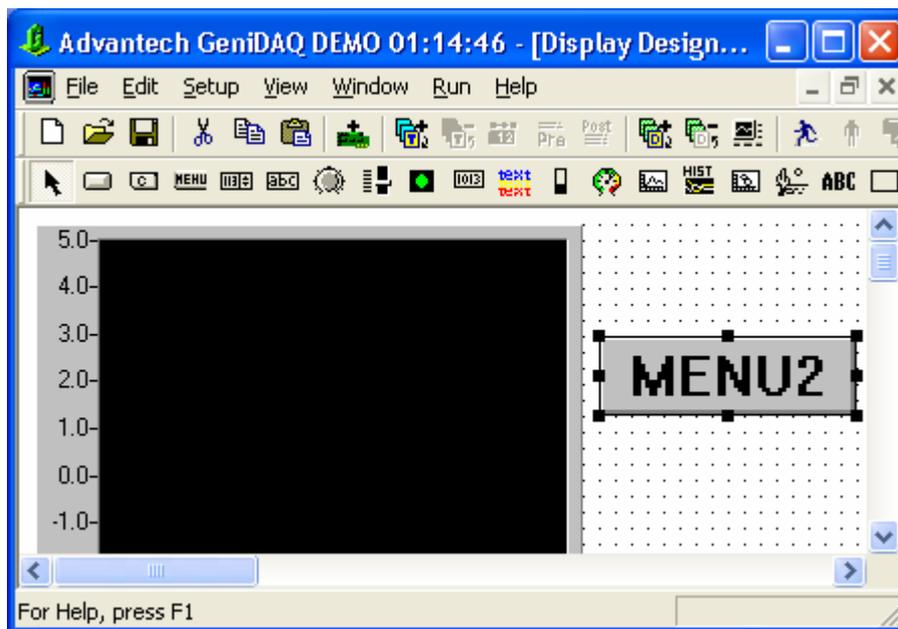


Рис. 4.24. Создание кнопки меню в окне DISP1

При передаче значений операндов блоку вычисления с одним оператором необходимо принимать ряд мер предосторожности. Операциям деления и вычисления остатка от целочисленного деления в качестве второго аргумента не должно передаваться нулевое значение. Кроме того, некоторым операторам (sqrt, ln, log) в качестве аргумента должно передаваться только положительное значение. Если не следовать указанным правилам, то при исполнении стратегии будут возникать неустраняемые ошибки.

Представленный в таблице оператор jct предназначен для выполнения специальной функции, которая заключается в передаче на выход блока значения, поступающего на его вход. Данная функция наиболее удобна в случае, если требуется установить связь между элементом управления *кнопка* (Binary Button) редактора форм отображения и несколькими функциональными блоками стратегии. Элемент управления *кнопка* может быть связан только с одним функциональным блоком стратегии. Оператор jct, вход которого связан с элементом управления *кнопка*, может быть использован в качестве "точки ветвления" для установления связи с любым количеством функциональных блоков стратегии. Блок вычисления с одним оператором имеет два входа — **Операнд 1** и **Операнд 2** — и один выход, по которому выводится результат выполнения математической операции, выбранной в

поле **Operator** диалогового окна настройки параметров блока. Справочную информацию о параметрах настройки диалогового окна, показанного на рис. 4.28, можно получить также с помощью кнопки **Help**. Конкретные настройки параметров блока вычислений с единственным параметром рассмотрены в примере, следующем далее.

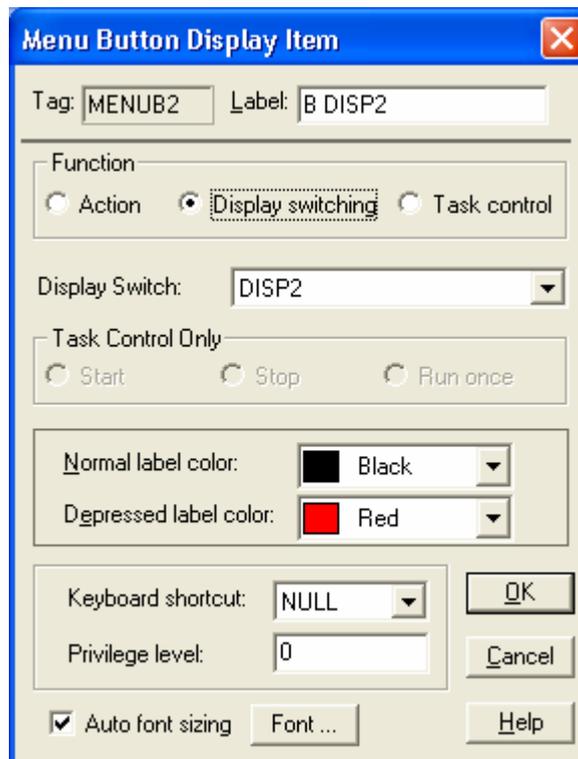


Рис. 4.25. Конфигурирование кнопки меню в окне DISP1

Для просмотра и изменения порядка выполнения функциональных блоков стратегии предназначены команды **View | Order Layout**, **Lauout | Complete Reorder** и **Layout | Exchange Order**. Их использование для просмотра и изменения порядка выполнения функциональных блоков также рассмотрено в примере, следующем далее.

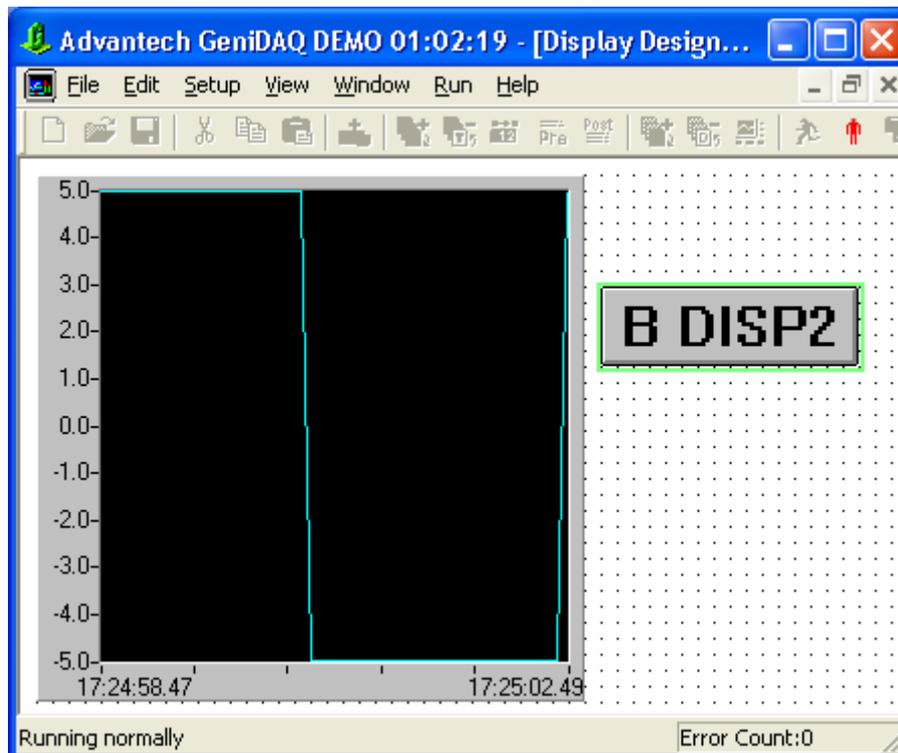


Рис. 4.26. Запуск проекта

4.3.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.29.
2. Выполните *второе действие* из занятия 1. В нашем примере также задан период запуска приложения 100 миллисекунд.
3. *Третье действие.* В окне редактора задач добавьте два блока аналогового ввода с позиционными обозначениями **AI1**, **AI2** соответственно и блок сложения **Single Calculation** (рис. 4.30). Настройте блоки аналогового ввода в соответствии с рис. 4.31 и 4.32.
4. *Четвертое действие.* С помощью проводника установите связь между блоком аналогового ввода **AI1** и блоком сложения. Для этого поместите курсор на пиктограмму проводника на панели инструментов редактора задач и произведите щелчок левой кнопкой мыши. Курсор мыши при этом примет вид катушки с нитками. Поместите курсор в область отображения блока **AI1** и произведите

щелчок левой кнопкой мыши. В появившемся окне диалога (рис. 4.33) выберите канал **Cannel0** и нажмите кнопку **ОК**. Переместите курсор по горизонтали в позицию, расположенную над блоком суммирования, щелкните левой кнопкой мыши, затем поместите курсор в область отображения блока суммирования и еще раз щелкните левой кнопкой мыши. В появившемся окне диалога (рис. 4.34) выберите **Operand1** и нажмите кнопку **ОК**. Аналогичным образом установите связь между блоком аналогового ввода **AI2** и блоком суммирования (выберите соответственно **Cannel0** и **Operand2**). В результате желаемые связи будут установлены (рис. 4.35), а блок сложения настроен нужным образом. Чтобы посмотреть настройку блока сложения, достаточно дважды щелкнуть левой кнопкой мыши над его изображением в окне редактора задач (рис. 4.36).

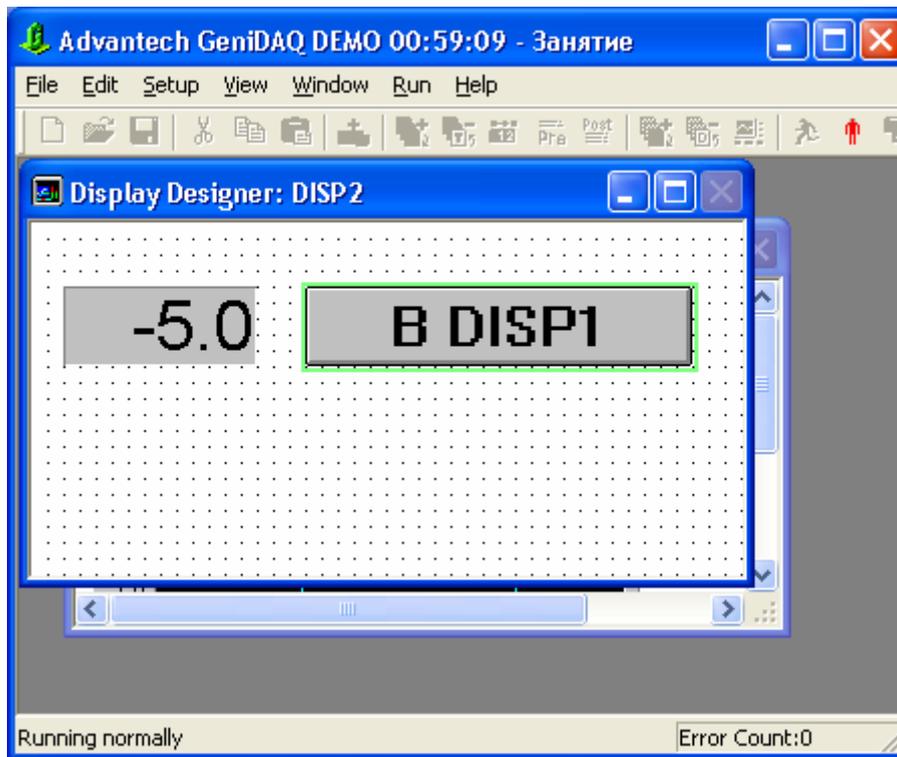


Рис. 4.27. Работа проекта после переключения окна отображения

5. *Пятое действие.* Аналогичным образом установите в окне редактора задач блок вычитания и установите связи между блоками аналогового ввода и блоком вычитания. Результирующий вид окна редактора задач показан на рис. 4.37, а настройка блока вычитания — на рис. 4.38.
6. *Шестое действие.* Сконфигурируйте график времени в соответствии с рис. 4.39 аналогично указанному в четвертом действии из занятия 1.

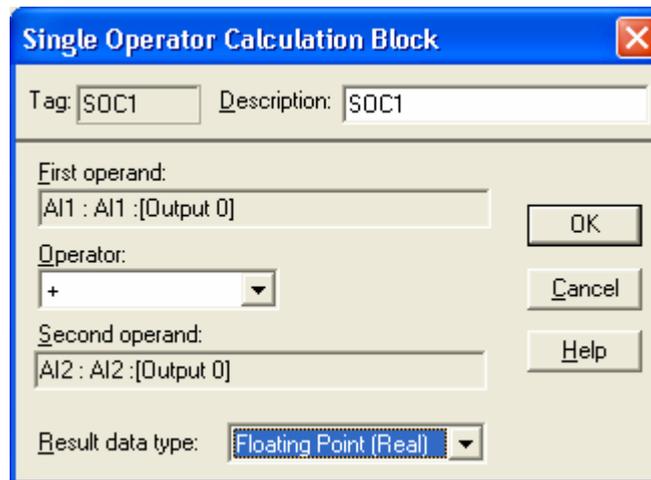


Рис. 4.28. Диалоговое окно настройки блока вычислений с одним оператором

Таблица 4.1. Операторы и функции вычисления с единственным оператором

Оператор	Результат на выходе блока
por	Всегда 0
+	<i>Операнд1 + Операнд2</i>
-	<i>Операнд1 - Операнд2</i>
x	<i>Операнд1 * Операнд2</i>
/	<i>Операнд1 / Операнд2</i>
pow	<i>Операнд1 в степени Операнд2</i>
mod*	Остаток целочисленного деления <i>Операнда1</i> на <i>Операнд2</i>
and*	Логическое И операндов
or*	Логическое ИЛИ операндов
xor*	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ операндов
max	Максимальное значение из двух операндов
min	Минимальное значение из двух операндов
>=	1, если <i>Операнд1</i> >= <i>Операнд2</i> ; 0 — в противном случае

Оператор	Результат на выходе блока
<=	1, если <i>Операнд1</i> <= <i>Операнд2</i> ; 0 — в противном случае
>	1, если <i>Операнд1</i> > <i>Операнд2</i> ; 0 — в противном случае
<	1, если <i>Операнд1</i> < <i>Операнд2</i> ; 0 — в противном случае
equ	1, если <i>Операнд1</i> равен <i>Операнду2</i> ; 0 — в противном случае
neq	1, если <i>Операнд1</i> не равен <i>Операнд2</i> ; 0 — в противном случае
abs	Абсолютная величина (модуль) <i>Операнда1</i>
not*	Логическое НЕ <i>Операнда1</i>
inv	Инверсия <i>Операнда1</i>
sqrt	Квадратный корень из <i>Операнда1</i>
log	Десятичный логарифм <i>Операнда1</i>
ln	Натуральный логарифм <i>Операнда1</i>
exp	$exp(Операнд1)$
jct	Оператор объединения (описание приведено ниже)

7. *Седьмое действие.* Сохраните созданный проект (стратегию) аналогично пятому действию из занятия 1.
8. *Восьмое действие.* После сохранения файлов созданного проекта имеется возможность немедленного запуска созданного проекта с помощью кнопки **Start** на панели инструментов (рис. 4.40). Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
9. *Девятое действие.* Посмотрите порядок выполнения функциональных блоков стратегии. Для этого включите команду **View | Order Layout** и на пиктограммах блоков аналогового ввода **AI1** и **AI2** будут показаны номера, определяющие порядок исполнения блоков в стратегии (рис. 4.41).

Для изменения порядка выполнения функциональных блоков стратегии выключите команду **View | Order Layout** и включите команду **Layout | Complete Reorder**. В результате этого в окне редактора задач появятся номера, определяющие порядок выполнения всех блоков стратегии, но в этом режиме можно изменить порядок выполнения только функциональных блоков. Для этого щелкните левой кнопкой мыши вначале над изображением блока **AI2**, а потом — над изображением блока **AI1** (рис. 4.42). Чтобы изменить порядок выполнения всех блоков стратегии выключите команду **Layout | Complete Reorder** и включите команду **Layout | Exchange Reorder**. Установите выполнение блоков в следующей последовательности: **AI2**, **AI1**, **SOC2** и **SOC1**. Для этого последовательно

щелкните левой кнопкой мыши над изображением блоков **SOC1** и **SOC2** (рис. 4.43). Выключите команду **Layout | Exchange Reorder**.

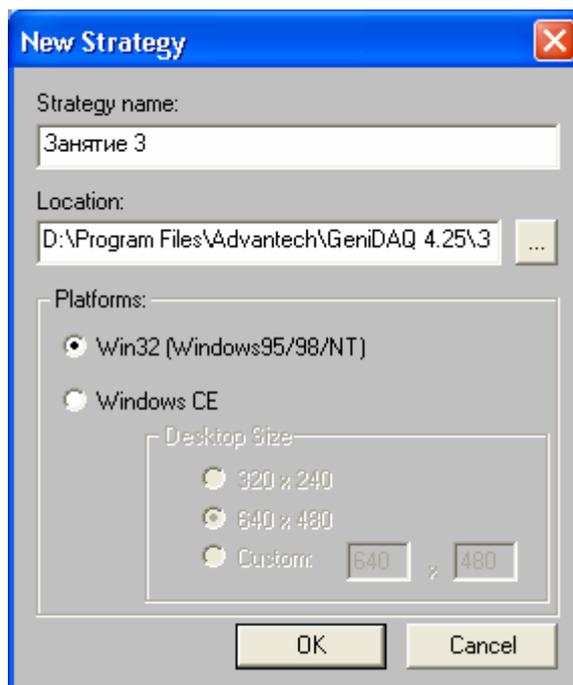


Рис. 4.29. Конфигурирование параметров стратегии

Запустите модифицированный проект после его сохранения с помощью кнопки **Start** на панели инструментов (рис. 4.44). Остановите работу проекта с помощью кнопки **Stop** на панели инструментов.

10. *Десятое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

4.3.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.8. Повтор создания и тестирования приложения "Занятие 3"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 3**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

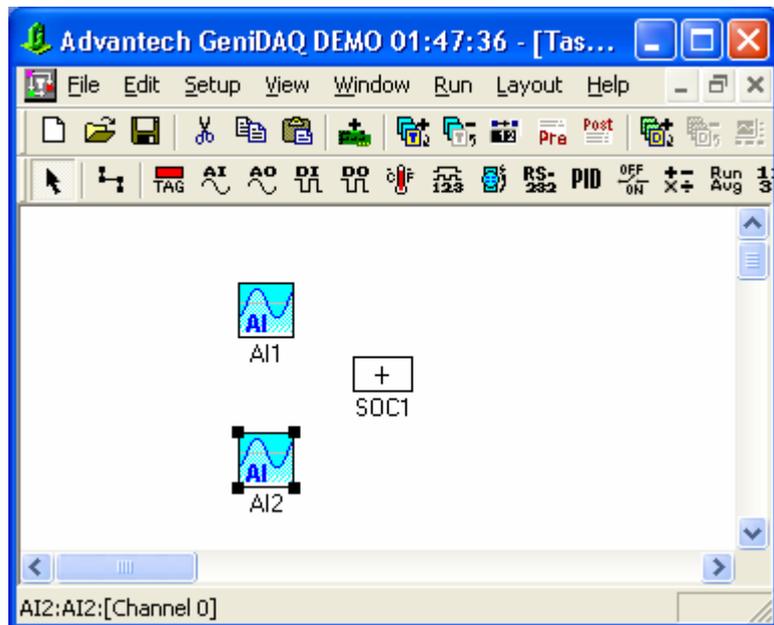


Рис. 4.30. Окно редактора задач с блоками аналогового ввода и блоком сложения

Упражнение 4.9

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 0.2 секунды. С помощью эмулятора и двух блоков аналогового ввода, помещенных в окно редактора задач, смоделируйте соответственно гармонический и случайный сигнал в десять раз и сложите его с гармоническим сигналом. В окне редактора форм отображения покажите указанные процессы на общем графике времени и с помощью цифровых индикаторов. Снабдите элементы отображения поясняющими надписями. Измените порядок выполнения функциональных блоков в окне редактора задач и наблюдайте вызванные этим изменения.

4.4. Занятие 4 "Использование инструментов рисования редактора форм отображения"

Цель занятия состоит в изучении правил применения инструментов рисования для создания пользовательских элементов отображения. Используются инструменты рисования и кнопка с двумя состояниями, имеющиеся на панели инструментов редактора задач. С помощью инструментов рисования создается некоторый графический объект. С помощью кнопки с двумя состояниями созданный объект изменяет свой цвет (элемент анимации).

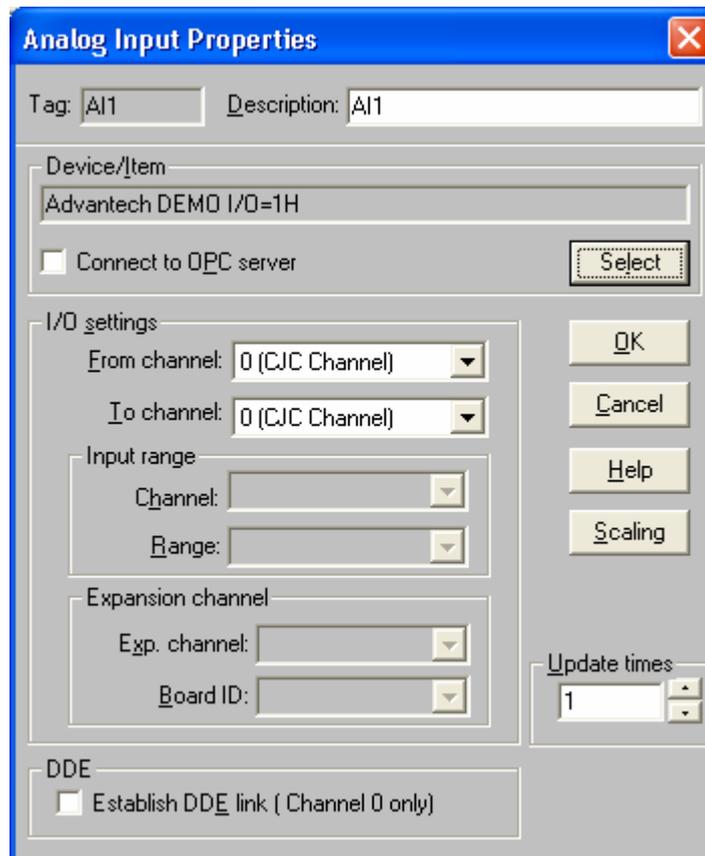


Рис. 4.31. Конфигурация блока аналогового АИ

4.4.1. Используемый инструментарий

Новым на данном занятии является использование в окне отображения кнопки с двумя состояниями и инструментов рисования (эллипс, прямоугольник) для создания графического объекта и его анимации.

Кнопка с двумя состояниями (*Binary Button*) может быть с помощью функционального блока *tag* (*Tag*) связана и использована для управления логическим состоянием любой двоичной или целочисленной переменной стратегии. Она позволяет передавать свое состояние из формы отображения в задачу. При нажатии кнопки с двумя состояниями производится передача логической 1 (или логического 0) связанному с ней блоку *tag*. Активизировать кнопку с двумя состояниями (нажать на кнопку) можно следующими способами:

- поместить курсор мыши на изображение кнопки в окне отображения и произвести щелчок левой клавишей мыши;

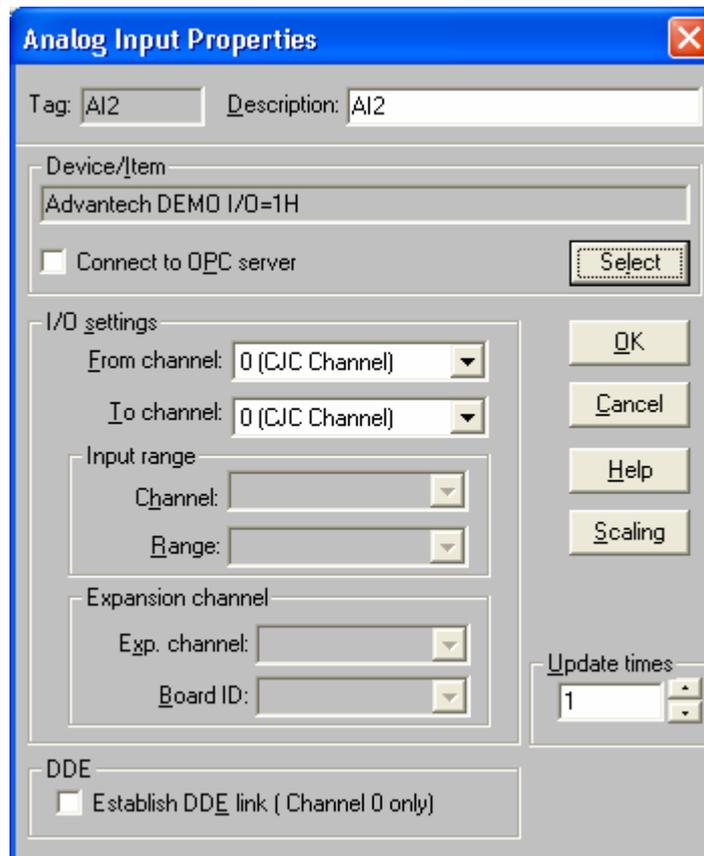


Рис. 4.32. Конфигурация блока аналогового AI2

- ❑ с помощью клавиши **Tab** на клавиатуре перевести фокус ввода на кнопку и нажать клавишу **Enter**;
- ❑ нажать клавишу, выбранную в комбинированном списке **Keyboard mapping** диалогового окна настройки параметров кнопки (рис. 4.45).

Размер и тип шрифта, используемого для отображения надписи на кнопке, могут быть установлены путем нажатия кнопки **Font** диалогового окна и выбора требуемого шрифта и его параметров из перечня шрифтов, зарегистрированных в ОС Windows. Поле *тег* (**Tag**) содержит идентификатор кнопки, который используется при установлении связи между кнопкой и другими элементами отображения либо функциональными блоками *тег* стратегии.

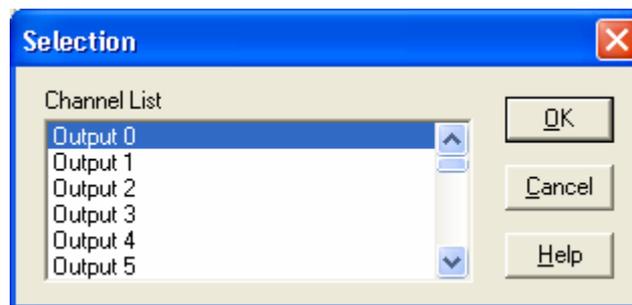


Рис. 4.33

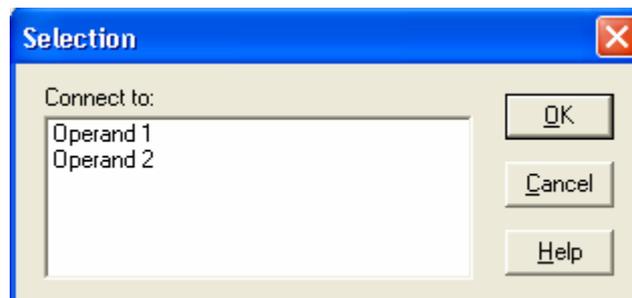


Рис. 4.34

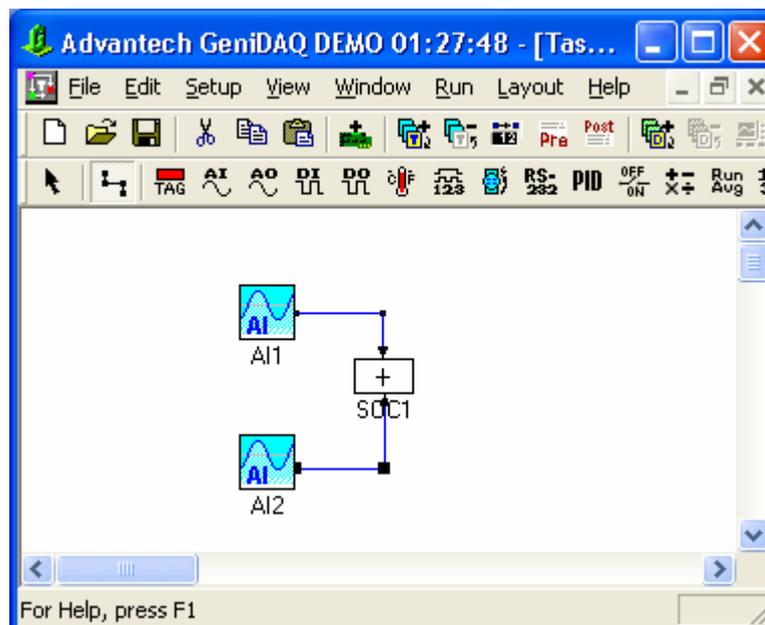


Рис. 4.35. Связь блоков аналогового ввода с блоком суммирования

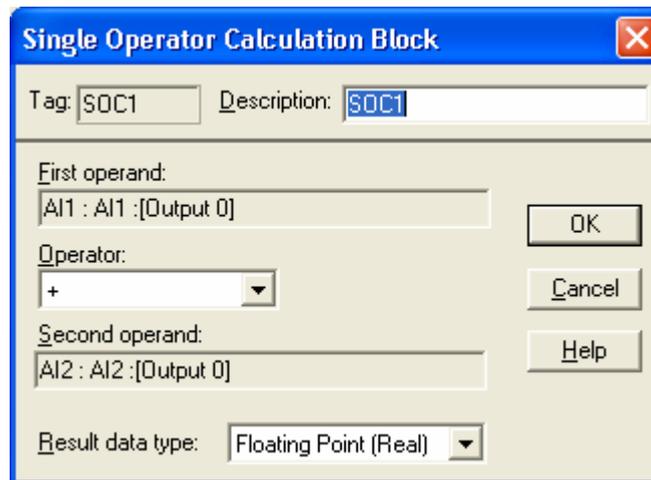


Рис. 4.36. Результирующая настройка блока суммирования

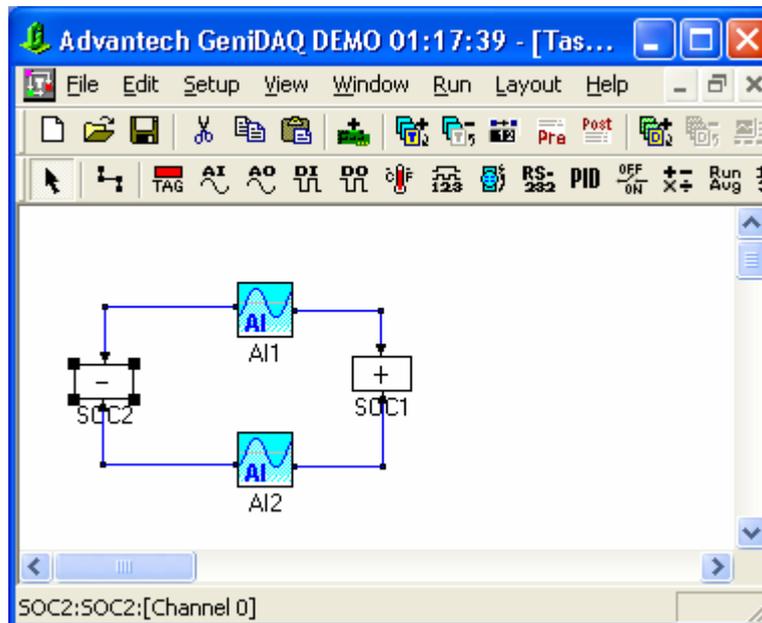


Рис. 4.37. Окно редактора задач с блоками аналогового ввода и блоками сложения и вычитания

Обратите особое внимание на то, что связь между кнопкой и любыми функциональными блоками в задачах должна осуществляться через функциональный блок тег редактора задач. Поле надпись на кнопке (**Label**) предназначено для ввода надписи, которая будет отображаться на кнопке в процессе исполнения стратегии.

Максимальное количество символов, из которых состоит надпись, составляет 30. Поле *режим функционирования* (**Operating style**) представлено комбинированным списком и используется для выбора режима функционирования кнопки.

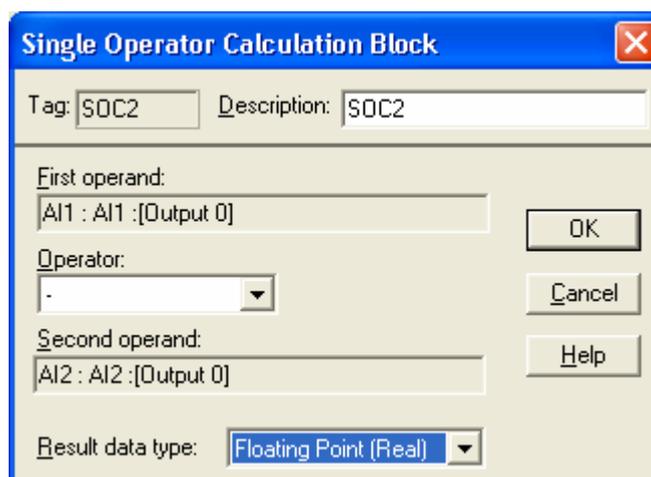


Рис. 4.38. Результирующая настройка блока вычитания

При выборе значения **On-Off** элемент управления будет выполнять функцию кнопки с фиксацией состояния после нажатия и отпускания с удержанием выходного значения для каждого состояния. При выборе значения **Monetary** элемент управления будет выполнять функцию кнопки без фиксации состояния (нажатие с мгновенным возвратом в отжатое состояние). Соответствующее значение на выходе удерживается только в течение времени, пока кнопка удерживается в нажатом состоянии. При выборе значения **Radio Button** элемент управления будет выполнять функцию кнопки, состояние которой зависит от состояния других связанных с ней кнопок. Данная кнопка носит название "радиокнопка". Поле *управление от клавиатуры* (**Keyboard mapping**), представленное комбинированным списком, позволяет выбрать клавишу на клавиатуре, с помощью которой возможно управлять состоянием кнопки при передаче ей фокуса ввода. Возможно использование функциональных клавиш от **F2** до **F9** или от **A** до **Z**. Поле *уровень привилегий* (**Privilege level**) предназначено для защиты функций управления положением коммутационных аппаратов, связанных с данной кнопкой. Уровень привилегий может принимать значения от 0 до 255, причем большему значению соответствует более высокий уровень привилегий. Таким образом, если для кнопки установлен уровень привилегий, равный 100, то нажать на данную кнопку смогут только пользователи с правами доступа от 100 и выше. Поля *цвет надписи на кнопке* (**Normal label color, Depressed label color**) позволяют установить цвета надписи на кнопке в нажатом и отпущенном состояниях. Реализована поддержка до 16 цветов. Переключатель *выходное значение* (**Output value**) предназначен для установки значений, передаваемых кнопкой другим элементам отображения и функциональным блокам *тег* стратегии при нажатии и отпускании.

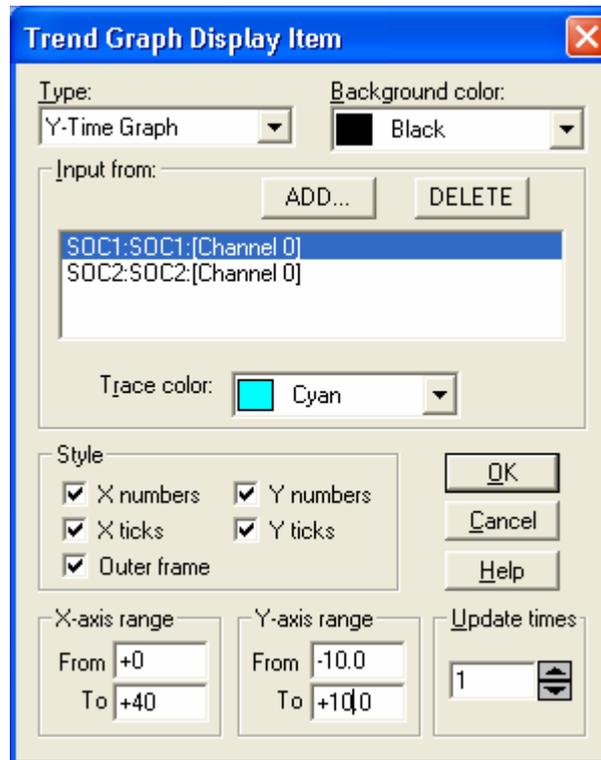


Рис. 4.39. Конфигурация графика времени

Имеется возможность передачи **0 в отжатом состоянии, 1 в нажатом состоянии** или **1 в отжатом состоянии, 0 в нажатом состоянии**. Независимый переключатель **звуковой сигнал при нажатии (Beep when pressed)** позволяет формировать звуковой сигнал при любом изменении состояния кнопки. Переключатель **автоподбор размера шрифта (Auto font sizing)** в отмеченном состоянии активизирует функцию автоматического выбора размера надписи при изменении размеров кнопки в процессе разработки стратегии.

Инструмент рисования "прямоугольник" предназначен для создания в окне экранной формы графического объекта прямоугольной формы, цвет которого определяется логическим состоянием на выходе присоединенного функционального блока стратегии (рис. 4.46).

Поле **Input from** содержит идентификатор выхода функционального блока, связанного с элементом отображения. Поле **Pen color** позволяет выбрать цвет рамки графического объекта. Возможен выбор одного из 16 цветов. Поле **Pen size** позволяет установить толщину рамки графического объекта.

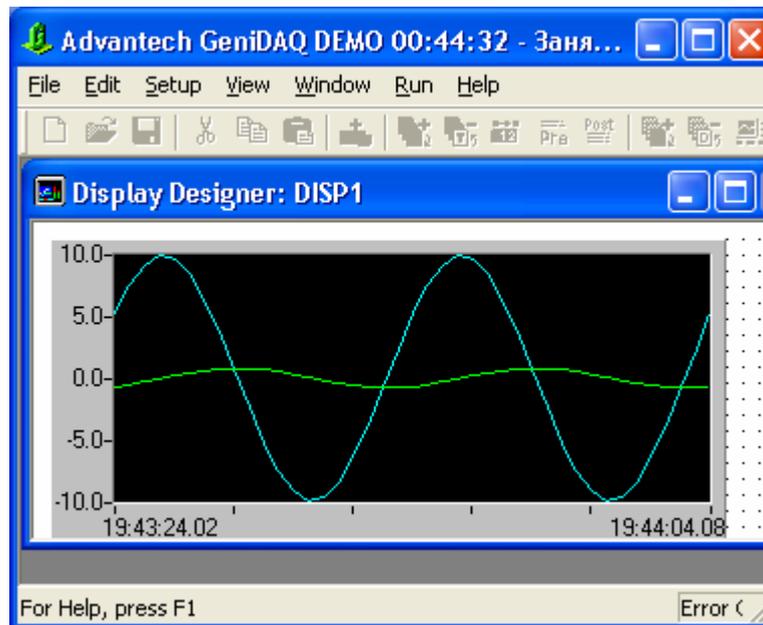


Рис. 4.40. Работа приложения сразу же после его запуска

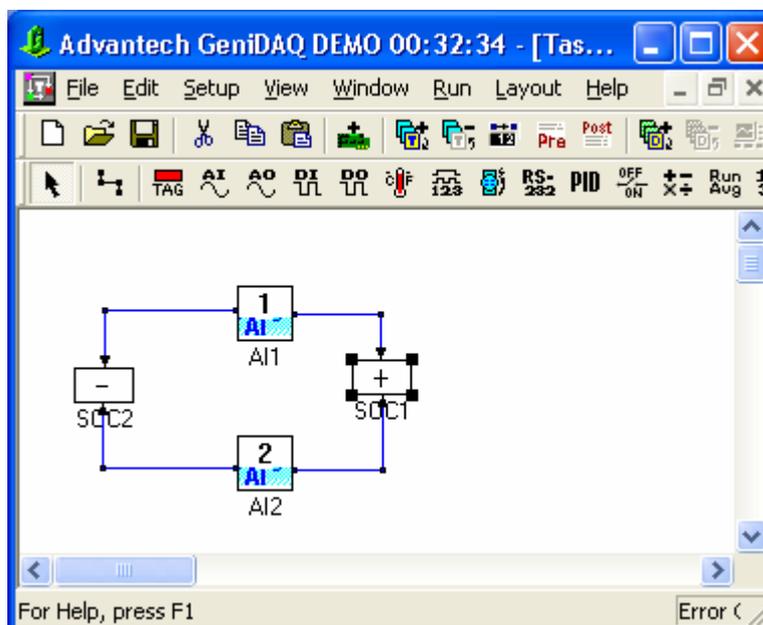


Рис. 4.41. Просмотр порядка выполнения функциональных блоков стратегии

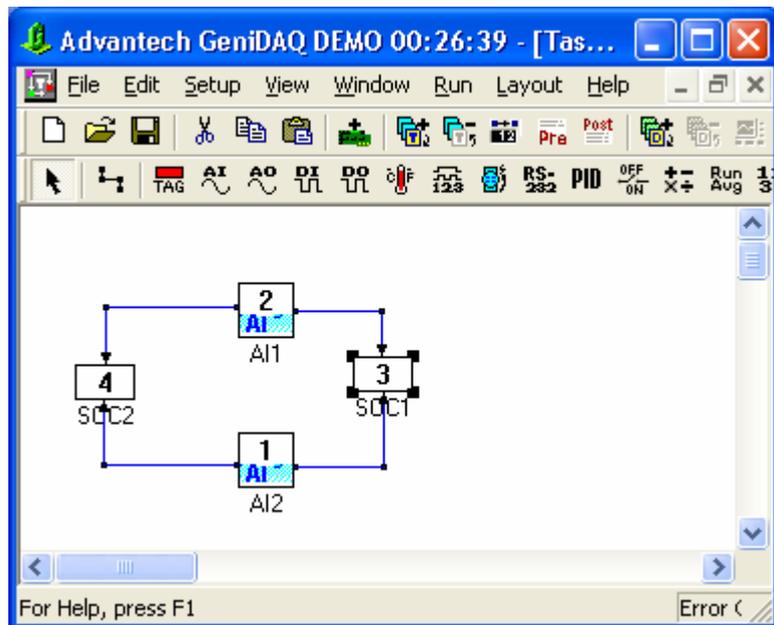


Рис. 4.42. Вид окна редактора задач после изменения порядка выполнения функциональных блоков

Толщина рамки задается в пикселях. Поле **Color when normal** позволяет выбрать цвет внутренней области графического объекта при поступлении на его вход логической единицы. Поле **Color when activated** позволяет выбрать цвет внутренней области графического объекта при поступлении на его вход логического нуля.

Инструмент рисования "эллипс". предназначен для создания в окне экранной формы графического объекта, имеющего форму эллипса, цвет которого определяется логическим состоянием на выходе присоединенного функционального блока стратегии (рис. 4.47).

Поле **Input from** содержит идентификатор выхода функционального блока, связанного с элементом отображения. Поле **Pen color** позволяет выбрать цвет рамки графического объекта. Возможен выбор одного из 16 цветов. Поле **Pen size** позволяет установить толщину рамки графического объекта. Толщина рамки задается в пикселях. Поле **Color when normal** позволяет выбрать цвет внутренней области графического объекта при поступлении на его вход логической единицы. Поле **Color when activated** позволяет выбрать цвет внутренней области графического объекта при поступлении на его вход логического нуля.

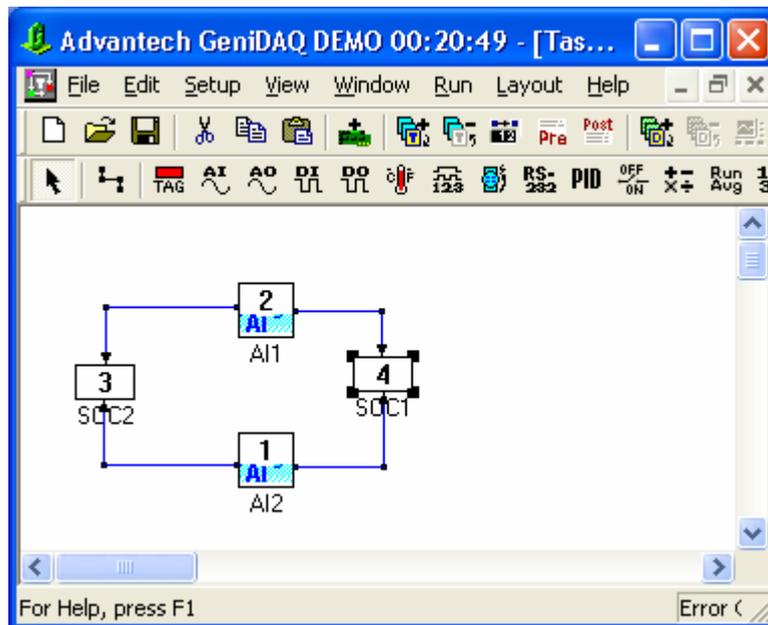


Рис. 4.43. Вид окна редактора задач после изменения порядка выполнения всех блоков стратегии

4.4.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.48.
2. *Второе действие.* Добавьте в окно редактора форм отображения кнопку с двумя состояниями **Binary Button**. Выполните над ней двойной щелчок левой кнопкой мыши и в появившемся диалоговом окне задайте параметры кнопки в соответствии с приведенным ранее рис. 4.45. Для сохранения параметров нажмите кнопку **ОК**.
3. *Третье действие.* Произведите щелчок левой кнопкой мыши над пиктограммой инструмента рисования *эллипс (Oval Draw)* на панели инструментов редактора форм отображения. Переместите курсор в окно отображения, нажмите левую кнопку мыши и, не отпуская ее, нарисуйте круг. С помощью команд **Edit | Copy** и **Edit | Paste** скопируйте созданный круг (он появится в левом верхнем углу окна отображения), уменьшите его размер. Для этого поместите курсор в угол рамки черного цвета и переместите курсор в направлении середины рамки. Поместите круг меньшего размера внутрь круга большего размера.

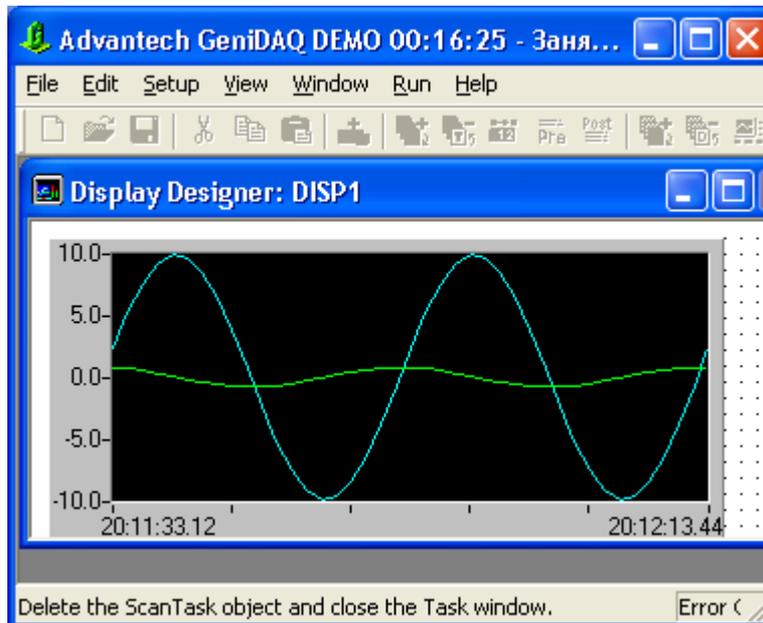


Рис. 4.44

Аналогичным образом создайте в окне отображения два графических примитива *прямоугольник* (**Rectangle Draw**). Расположите созданные графические примитивы таким образом, чтобы они образовали графическое изображение насоса. При необходимости используйте команды **Edit | Bring to Front** и **Edit | Send to Back**. Объедините графические примитивы один графический объект. Для этого выполните операцию переноса курсора для одновременного выделения всех примитивов, образующих графическое изображение насоса, и выполните команду **Edit | Make Object**. Окно отображения приобретет вид, показанный на рис. 4.49. Выполните двойной щелчок левой кнопкой мыши над изображением созданного графического объекта, в появившемся окне диалога нажмите кнопку **Select**, установите связь графического объекта с кнопкой с двумя состояниями в соответствии с рис. 4.50 и нажмите кнопку **OK**. Окно диалога приобретает вид, показанный на рис. 4.51. Нажатием кнопки **OK** закройте окно диалога.

4. *Четвертое действие.* Сохраните созданный проект (стратегию) аналогично тому, как это делалось на занятии 1. Запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.52). Понажимайте на кнопку с двумя состояниями — цвет графического объекта будет меняться при каждом нажатии на кнопку. Остановите работу проекта с помощью кнопки **Stop** на панели инструментов.
5. *Пятое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

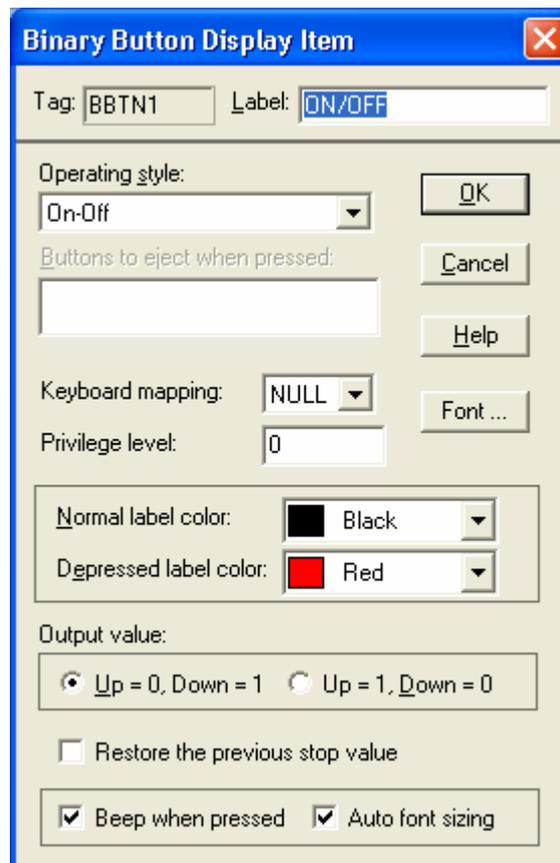


Рис. 4.45. Диалоговое окно конфигурирования кнопки с двумя состояниями

Замечание

Обратите внимание на то, что в частном случае окно редактора задач может не использоваться. Именно такая ситуация имела место в приложении **Занятие 4**.

4.4.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.10. Повтор создания и тестирования приложения "Занятие 4"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 4**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

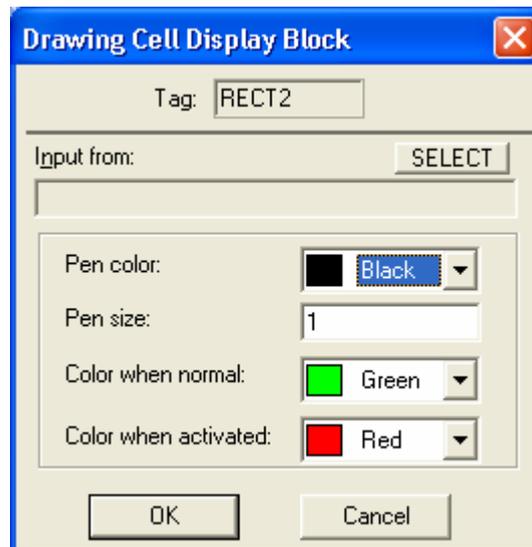


Рис. 4.46. Окно конфигурирования графического примитива Прямоугольник

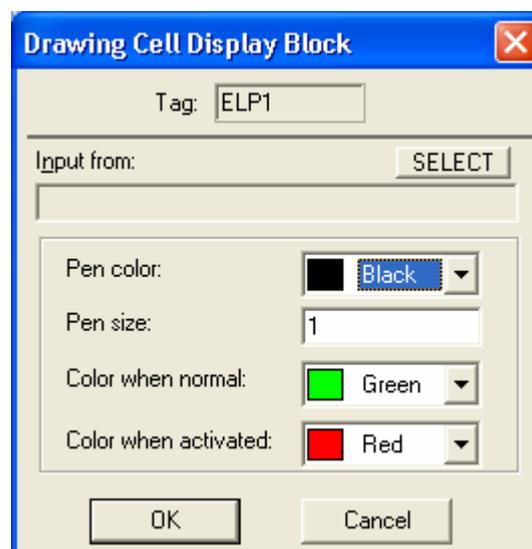


Рис. 4.47. Окно конфигурирования графического примитива Эллипс

4.5. Занятие 5 "Использование тега для связи между задачей и формой отображения"

Цель занятия состоит в изучении правил использования функционального блока *тег* (**Tag**) редактора задач для получения данных, вводимых пользователем с помощью какого либо элемента управления, используемого в окне отображения. Используются функциональные блок *тег* (**Tag**), блок *архива тревог* (**Alarm Log**) редактора задач и инструменты *инкрементный регулятор* (**Numeric Control**), *индикатор* (**Indicator**) и *текстовые строки* (**Text String**) редактора форм отображения.

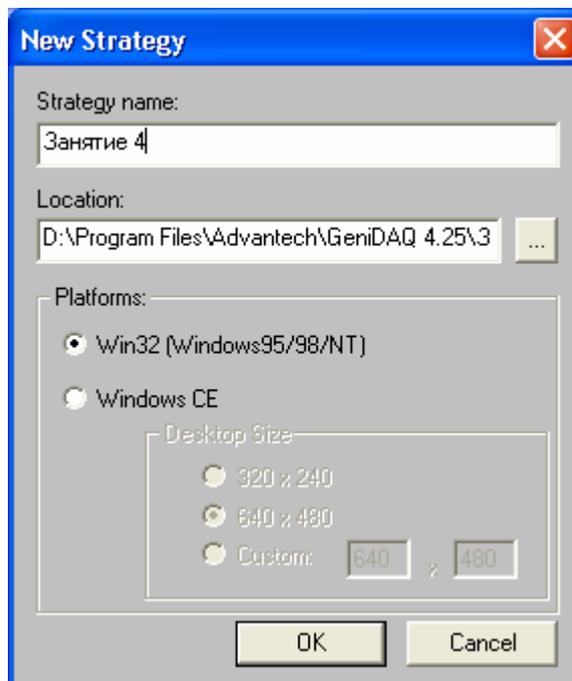


Рис. 4.48. Конфигурирование параметров стратегии

4.5.1. Используемый инструментарий

Для построения пользовательского приложения, выполняющего требуемые функции, используются перечисленные ранее функциональные блоки редактора задач и инструменты редактора форм отображения. Дадим их краткую характеристику.

Функциональный блок *тег* (**Tag**) редактора задач предназначен для установления связи между элементами управления редактора форм отображения с одной стороны и функциональными блоками редактора задач или виртуальными тегами с другой стороны (рис. 4.53). На данном занятии тег используется для связи управляющего элемента из окна отображения с функциональным блоком редактора задач.

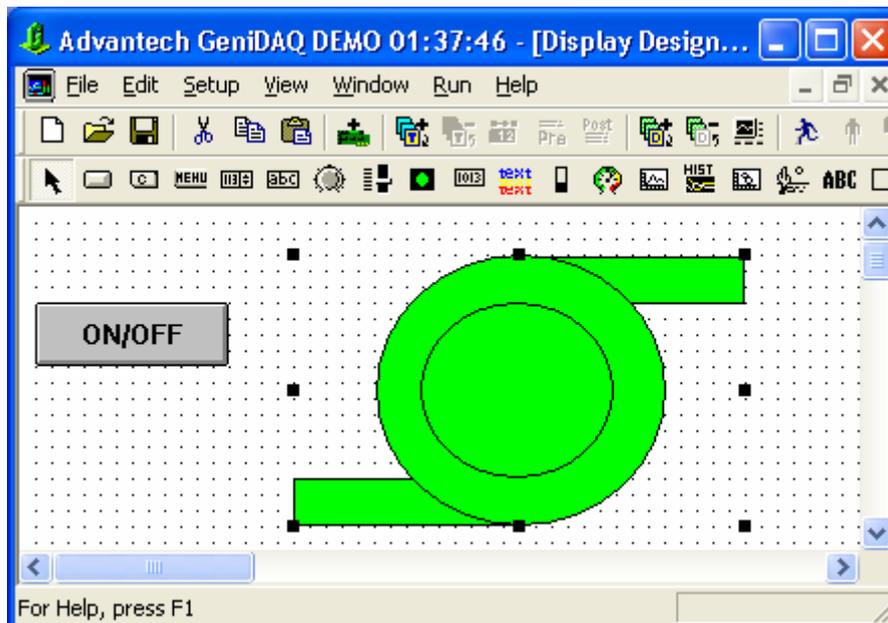


Рис. 4.49. Вид окна отображения после создания графического объекта (насоса)



Рис. 4.50. Установление связи графического объекта с кнопкой

Блок *tag* имеет непосредственный доступ к значениям элемента управления редактора форм отображения или виртуального тега центра обработки данных, указанного в полях группы **Attaching to**. На рис. 4.53 таким элементом является управляющий элемент **NCTL1** окна отображения. Получаемые значения *tag* передает функциональному блоку окна редактора задач, присоединенному к его выходу.

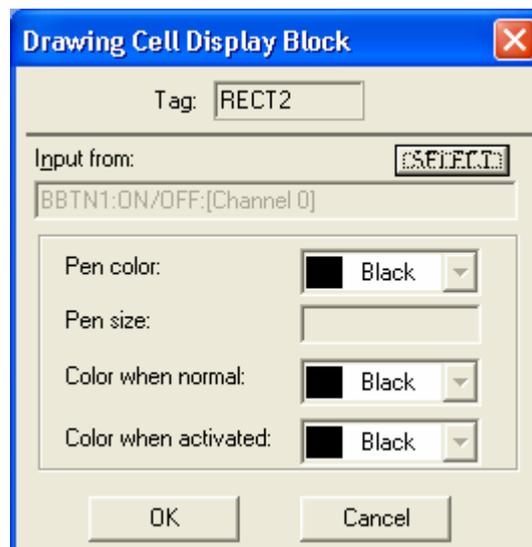


Рис. 4.51. Конфигурирование параметров графического объекта

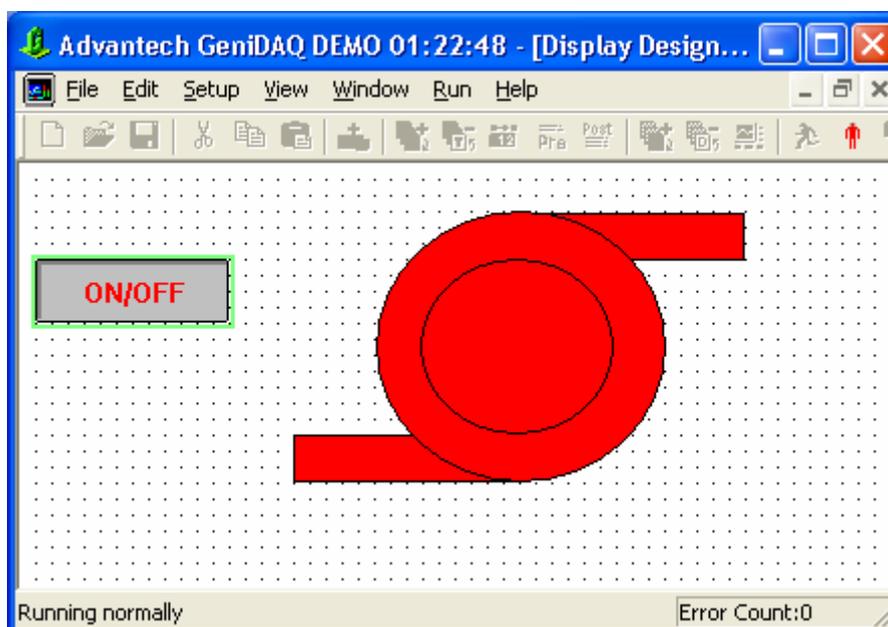


Рис. 4.52. Работа приложения

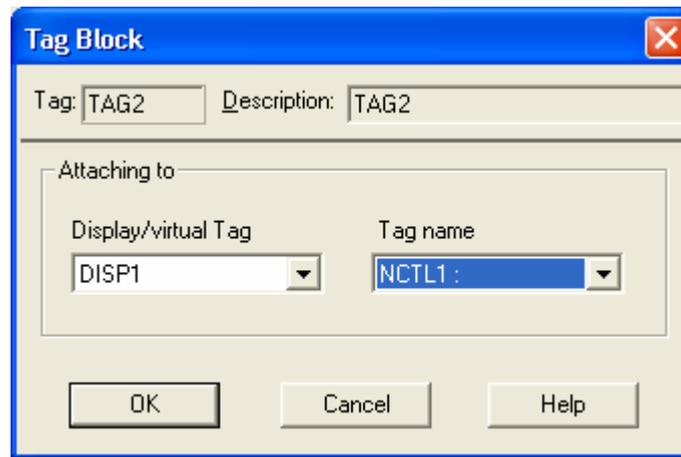


Рис. 4.53. Конфигурирование функционального блока *тег*

Функциональный блок архива тревог (*Alarm Log*) редактора задач предназначен для сохранения в архиве информации о зафиксированных аварийных событиях, связанных с сигналом, поступающим на вход блока архива тревог. Блок имеет вход и выход. Тревоги можно фиксировать в файле архива событий (..\GeniDAQ 4.25\genieDAQ.elf). Для включения такого режима необходимо выполнить команду **Setup | Runtime Preference...** и в появившемся окне диалога установить флаг **Enable event log**. Сообщения об аварийных событиях могут в процессе исполнения стратегии отображаться в окне *журнала событий* (окно делается видимым при выполнении команды **View | Event Log**), когда значение на входе блока попадает в следующие диапазоны:

1. Выше верхнего предельного значения.
2. Между максимальным и верхним предельным значениями.
3. Между минимальным и нижним предельным значениями.
4. Ниже нижнего предельного значения.

На выходе блока присутствует целое число, соответствующее событию, зафиксированному блоком. Это позволяет представлять информацию об аварийных событиях в графической форме путем установления связи между блоком архива тревог и элементом отображения *индикатор*. Зависимость значений на выходе блока от зафиксированного блоком события приведена в табл. 4.4.

Таблица 4.2. Зависимость значения на выходе блока тревог от зафиксированного им события

Аварийное событие (значение на входе блока)	Значение на выходе блока
Выше верхнего предельного значения	4

Аварийное событие (значение на входе блока)	Значение на выходе блока
Между максимальным и верхним предельным значениями	2
Между максимальным и минимальным значениями	0
Между минимальным и нижним предельным значениями	1
Ниже нижнего предельного значения	3

Диалоговое окно конфигурирования блока архива тревог показано на рис. 4.54. При исполнении стратегии имеется возможность подтверждения восприятия оператором событий, зафиксированных блоком, путем двойного щелчка левой клавишей мыши в окне *журнала событий*. До подтверждения информация об аварийном событии отображается в окне *журнала событий* в виде строки красного цвета. Функциональный блок *архива тревог* имеет один вход, на который может поступать сигнал от другого функционального блока стратегии. Значение на входе проверяется блоком на вхождение в пределы, заданные в группе параметров *значения параметров тревоги (Alarm Settings)* диалогового окна настройки параметров блока в процессе разработки стратегии. На выходе блока присутствует целое число, соответствующее событию, зафиксированному блоком (см. табл. 4.2).

Инкрементный регулятор (Numeric Control) редактора форм отображения может быть помещен в окно отображения и связан с входной переменной функционального блока задачи, входящей в стратегию, и/или элементами отображения. Имеется возможность установки требуемых размеров регулятора. Инкрементный регулятор предназначен для ввода оператором числовых значений с помощью клавиатуры или мыши и передачи введенных значений связанному функциональному блоку стратегии, что позволяет реализовывать функции оперативного диспетчерского управления. Диалоговое окно конфигурирования инкрементного регулятора показано на рис. 4.55. Вводимые и отображаемые значения могут быть целого или вещественного типа (групповое окно **Data type**). Формат вводимого значения (количество цифр после десятичной точки — **Digits of precision**) может быть установлен только для вещественных чисел. Тип и размер используемого шрифта выбираются путем нажатия кнопки **Font...** Поле **Privilege level** предназначено для защиты функций управления, связанных с инкрементным регулятором. Уровень привилегий может принимать значения от 0 до 255, причем большему значению соответствует более высокий уровень привилегий. Таким образом, если для регулятора установлен уровень привилегий, равный 100, то изменить значение на его выходе смогут пользователи с правами доступа от 100 и выше. Поле **Initial value** определяет значение на выходе инкрементного регулятора при запуске стратегии на исполнение. Поле **Step value** определяет величину уменьшения/увеличения значения на выходе регулятора при однократном щелчке левой клавишей мыши на кнопках уменьшения/увеличения, расположенных справа от области отображения значения на выходе инкрементного регулятора. Поля **High limit/Low limit** предназначены для установки диапазона изменения значения на выходе инкрементного регулятора.

The image shows a software dialog box titled "Alarm Log Block". At the top, there are two text input fields: "Tag:" with the value "ALOG1" and "Description:" with the value "ALOG1". Below these is a section titled "Alarm settings" containing four numerical input fields: "High-High:" (90.0), "High:" (80.0), "Low:" (20.0), and "Low-Low:" (10.0). To the right of these fields are three buttons: "OK", "Cancel", and "Help". Below the "Alarm settings" section is another section titled "Alarm message format" which contains a list of checkboxes. The checked items are: "Date (MM/DD/YYYY)", "Time (HH:MM:SS)", "Alarm type (HI-HI, HI, LO, LO-LO)", "Tag name", and "Operator name (only the first 10 characters)". The unchecked items are: "Comment (30)", "Value", and "Limit value". A text input field is visible next to the "Comment (30)" checkbox.

Рис. 4.54. Конфигурирование функционального блока архива тревог

Индикатор (Indicator) редактора форм отображения представляет собой единичный индикатор, предназначенный для отображения логического состояния связанного с ним дискретного выхода функционального блока стратегии. Индикатор переводится в состояние *включено* при появлении логической единицы (ненулевое целое значение) на выходе связанного с ним функционального блока и в состояние *выключен* при появлении логического нуля (нулевое целое значение) на выходе связанного с ним функционального блока. Размеры, форма и цвет индикатора в разных состояниях могут быть установлены с помощью диалогового окна настройки параметров индикатора (рис. 4.56).

Поле **Input from** устанавливает связь индикатора с одной из переменных задачи, входящей в стратегию. Перед началом настройки параметров индикатора необходимо установить связь с функциональным блоком, логическое состояние на выходе которого предполагается отображать. Для этого в окне настройки индикатора следует нажать кнопку **SELECT** и в появившемся окне выбрать идентификатор задачи, идентификатор функционального блока и номер присоединяемого выхода функционального блока.

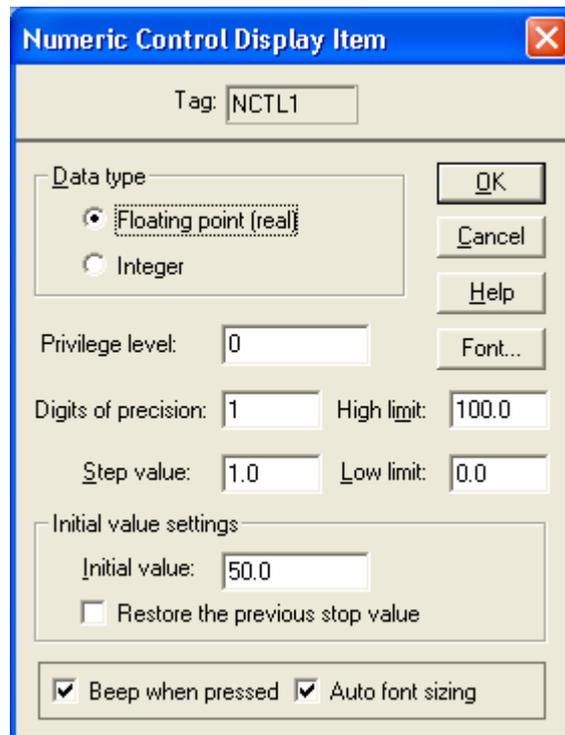


Рис. 4.55. Конфигурирование инкрементного регулятора



Рис. 4.56. Конфигурирование индикатора

При запуске стратегии на исполнение логическое состояние на выходе присоединенного функционального блока будет отображаться соответствующим цветом индикатора. Поле **Style** предназначено для выбора формы индикатора. Индикатор может иметь прямоугольную (квадратную) форму или форму эллипса (круга). Поля **Color for ON (1) state/Color for OFF (0) state** предназначены для

выбора цвета индикатора во включенном и выключенном состояниях. Реализована поддержка до 16 цветов.

Элемент отображения *Текстовая строка* (*Text string*) редактора форм отображения не имеет средств связи с функциональными блоками и другими элементами отображения/управления стратегии и предназначен для вывода на экран монитора статической символьной строки, которая определяется на этапе разработки стратегии. Имеется возможность выбора типа и размера шрифта, которым будет выводиться текстовая строка (рис. 4.57).

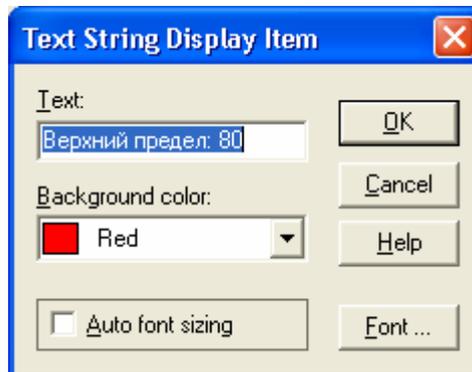


Рис. 4.57. Конфигурирование текстовой строки

Поле **Text** предназначено для ввода строки символов, которая будет выводиться во внутренней области элемента отображения во время исполнения стратегии. Поле **Background color** позволяет выбрать цвет внутренней области элемента отображения. Если требуется установить цвет и размер шрифта, следует снять отметку с флажка **Auto font sizing** и нажать кнопку **Font...** диалогового окна, после чего выбрать требуемый вид, размер и цвет в появившемся диалоговом окне настройки параметров шрифта.

4.5.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.57.
2. Выполните *второе действие* из занятия 1. В нашем примере также задан период запуска приложения 100 миллисекунд.
3. *Третье действие.* Добавьте в окно отображения инкрементный регулятор и настройте его в соответствии с рис. 4.55. Добавьте в окно отображения индикатор и настройте его в соответствии с рис. 4.56. Добавьте в это же окно три текстовых строки и настройте их в соответствии с рис. 4.57, 4.59 и 4.60. В результате окно отображения приобретает вид, показанный на рис. 4.61.

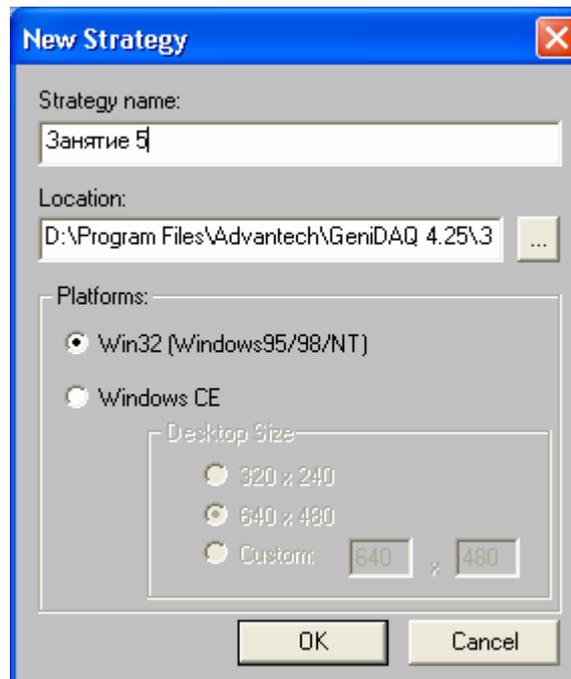


Рис. 4.58. Конфигурирование приложения

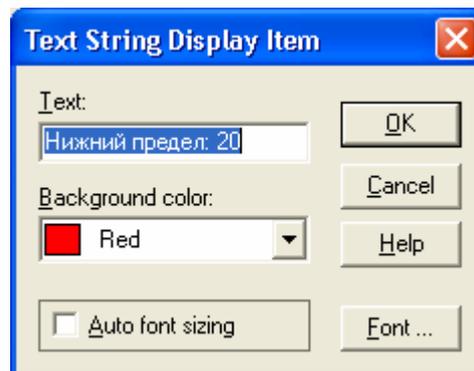


Рис. 4.59. Конфигурирование текстовой строки

4. *Четвертое действие.* Выдвиньте на передний план окно редактора задач и разместите в нем функциональные блоки *тег* и блок *архива тревог*. Произведите двойной щелчок левой кнопкой мыши над блоком *тег* и появится диалоговое окно его настройки (см. выше рис. 4.53). Сконфигурируйте блок *архива тревог* в соответствии с рис. 4.54. Соедините выход *тега* со входом блока *архива тревог*. Окно редактора задач приобретет вид, показанный на рис. 4.62.

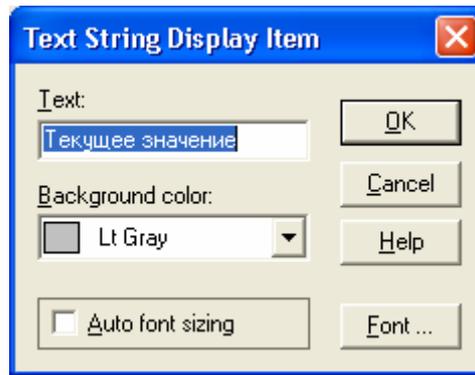


Рис. 4.60. Конфигурирование текстовой строки

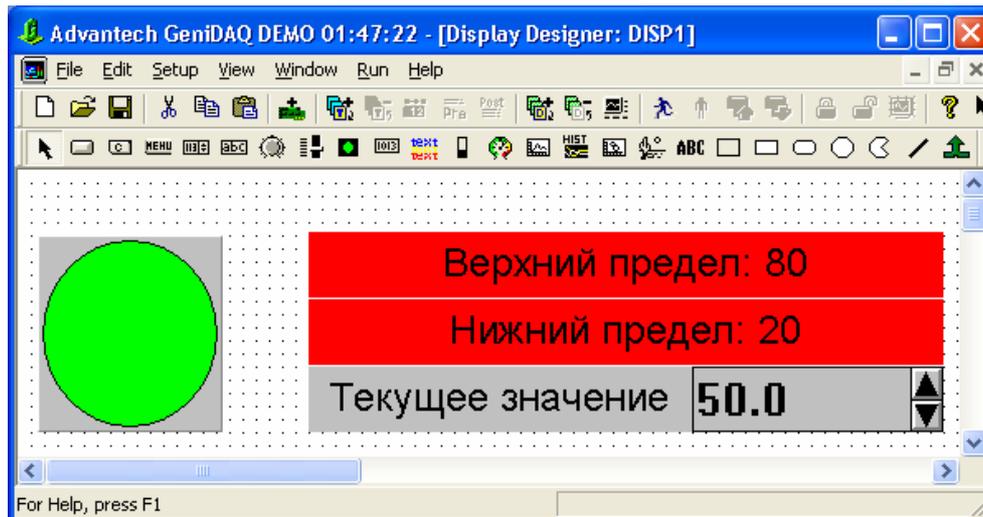


Рис. 4.61. Вид окна отображения

5. *Пятое действие.* Сохраните созданный проект (стратегию) аналогично пятому действию из занятия 1. Выполните команду **Setup | Runtime Preference...** и в появившемся окне диалога установите флаг **Enable event log**. Запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.63). Для отображения журнала событий выполните команду **View | Event Log** (рис. 4.64). Используя инкрементный регулятор задайте текущие значения 81 и 19 и обратите внимание на изменение состояния индикатора и связанного с этим появление звуковых сигналов. Подтвердите аварийные сообщения двойным щелчком левой клавиши мыши на строках журнала событий, отображаемых красным цветом. Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.

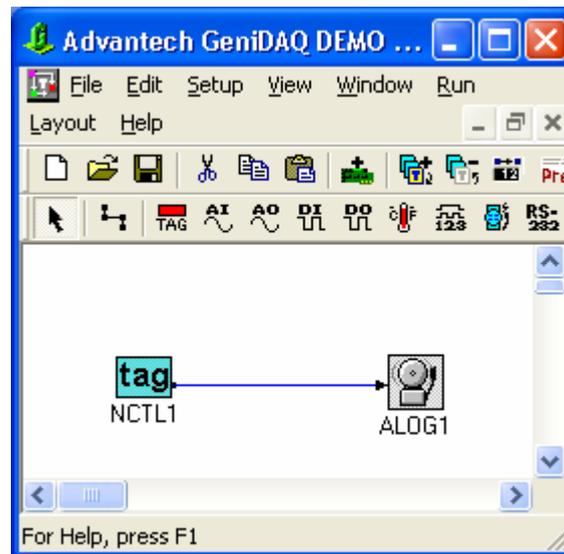


Рис. 4.62. Вид окна редактора задач

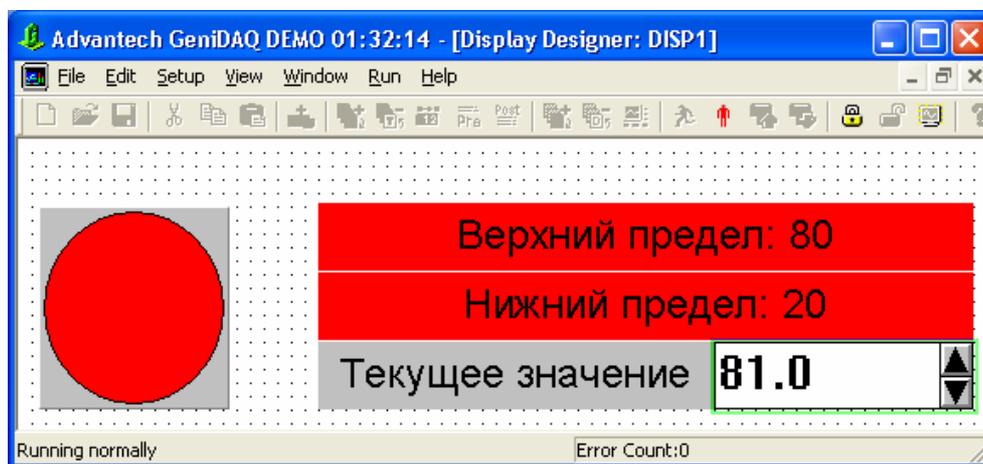


Рис. 4.63. Демонстрация работы проекта в окне отображения

6. *Шестое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.



Рис. 4.64. Вид журнала событий

4.5.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.11. Повтор создания и тестирования приложения "Занятие 5"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 5**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

Упражнение 4.12

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 0.2 секунды. С помощью эмулятора и блока аналогового ввода, помещенного в окно редактора задач, смоделируйте гармонический сигнал. Поместите в окно редактора задач функциональный блок тревог и подайте на него выходной сигнал блока аналогового ввода. В окне отображения с помощью индикатора отобразите состояние гармонического сигнала (для диапазона High-Low зеленым цветом, а в остальных случаях — красным). Подключите журнал событий и подтвердите получение аварийных сигналов. Снабдите элементы отображения поясняющими надписями.

Упражнение 4.13

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 1 секунда. Сконфигурируйте окно отображения в соответствии с рис. 4.65. В окне редактора задач используйте блок архива тревог и сконструируйте окно таким образом, чтобы в процессе работы обеспечить следующую последовательность активизации индикаторов: 3-2-1-2-3-4-5-4-3 и т. д.

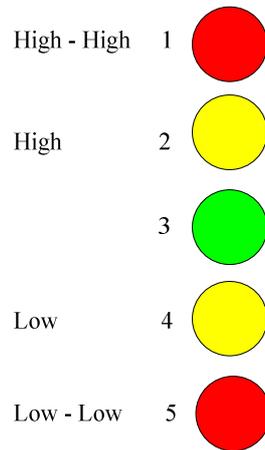


Рис. 4.65. Вид окна отображения (цвета индикаторов: 1, 5 — красный, 2, 4 — желтый, 3 — зеленый, альтернативный цвет — белый)

Упражнение 4.14

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 1 секунда. Сконфигурируйте окно отображения в соответствии с рис. 4.65. В окне редактора задач используйте блок архива тревог и сконструируйте окно таким образом, чтобы в процессе работы обеспечить следующую последовательность активизации индикаторов: 3-4-5-4-3-2-1-2-3 и т. д.

Упражнение 4.15

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 1 секунда. Сконфигурируйте окно отображения в соответствии с рис. 4.66. В окне редактора задач используйте блок архива тревог и сконструируйте окно таким образом, чтобы в процессе работы обеспечить следующую последовательность активизации индикаторов: 2-1-2-3-2-1 и т. д.

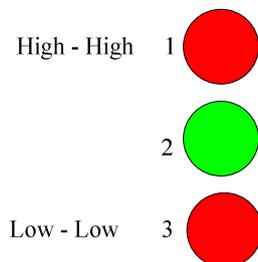


Рис. 4.66. Вид окна отображения (цвета индикаторов: 1, 3 — красный, 2 — зеленый, альтернативный цвет — белый)

Упражнение 4.16

Разработайте стратегию, обладающую следующими возможностями. Задайте период запуска задачи 1 секунда. Сконфигурируйте окно отображения в соответствии с рис. 4.66. В окне редактора задач используйте блок архива тревог и сконструируйте окно таким образом, чтобы в процессе работы обеспечить следующую последовательность активизации индикаторов: 2-3-2-1-2-3 и т. д.

Замечание

Рассматриваемые далее учебные примеры используют программы, написанные на языке Visual Basic for Applications (VBA). Поэтому следующий раздел главы посвящен рассмотрению языка VBA.

4.6. Использование языка VBA в SCADA-системах

Язык VBA широко применяется в SCADA-системах в качестве инструментального средства программирования. Далее последовательно приводится краткое описание основных средств VBA, не зависящих от среды его использования [14, 15], и рассматриваются средства привязки языка VBA к SCADA-системе GeniDAQ (фирма Advantech, США).

4.6.1. Средства языка VBA, инвариантные к среде использования (Microsoft Visual Basic 6.3)

Язык VBA, в отличие от языка Visual Basic (VB), не является языком объектно-ориентированного программирования в строгом смысле этого слова, но в нем широко используются элементы объектно-ориентированного подхода и связанные с ним понятия. VBA — подмножество языка VB, которое включает почти все средства создания приложений, структуры данных, управляющие структуры и возможность создания пользовательских типов данных.

Язык VBA является языком визуального и событийно управляемого программирования. Он позволяет создавать нестандартные диалоговые окна с набором элементов управления, процедуры, обрабатывающие события, возникающие при тех или иных действиях системы и пользователя. Проект (программа) на языке VBA, в отличие от программ на других языках, — результат побочной деятельности по созданию некоторого объекта (документа). Более того, проект VBA нельзя создать независимо от какого-либо объекта, даже если никакие свойства этого объекта не используются.

Язык VBA имеет графическую интегрированную среду разработки и реализует концепцию визуального программирования, управляемого событиями. Одной из важных областей применения языка VBA является его использование для программирования алгоритмов управления и других целей в SCADA-системах. Применительно к этому и ведется дальнейшее изложение материала.

4.6.1.1. Типы данных

В языке VBA имеются следующие типы данных: **Byte** (байт), **Boolean** (логический), **Integer** (целое), **Long** (длинное целое), **Single** (с плавающей точкой обычной точности), **Double** (с плавающей точкой двойной точности), **Currency** (денежный), **Decimal** (масштабируемое целое), **Date** (даты и время), **Object** (объект), **String** (строка), **Variant** (тип используется по умолчанию) — тип данных, определяемый пользователем, а также специальные типы объектов (табл. 4.3).

Таблица 4.3. Типы данных языка VBA

Тип	Размер в байтах	Диапазон значений
Boolean	2	True или False
Byte	1	От 0 до 255 (беззнаковое)
Date	8	От 0:00:00 1 января 0100 г. до 23:59:59 31 декабря 9999 г.
Currency	8	Характеризует денежную величину и принимает значения от 922 337 203 685 477.580 8 до -922 337 203 685 477.580 7
Decimal	12	От 0 до +/-79 228 162 514 264 337 593 543 950 335 без десятичной точки. От 0 до +/-7.922 816 251 426 433 759 354 395 033 5 с 28 знаками справа от десятичной точки; наименьшее ненулевое число +/-0.00000000000000000000000000000001 (+/-1E-28)
Double (double precision floating point)	8	От -1.797 693 134 862 315 70E+308 до 4.940 656 458 412 465 44E-324 для отрицательных значений. От 4.940 656 458 412 465 44E-324 до 1.797 693 134 862 315 70E+308 для положительных значений
Integer	2	От -32 768 до 32 767
Long (long integer)	4	От -2 147 483 648 до 2 147 483 647
Object	4	Переменная объектного типа, содержащая ссылку на объект (объект может быть целого типа)
Single (single precision floating point)	4	От -3.402 823 5E+38 до -1.401 298E-45 для отрицательных значений. От 1.401 298E-45 до 3.402 823 5E+38 для положительных значений
String (variable length)	Реализация зависит от платформы	От 0 примерно до 2 млрд. двухбайтовых юникод-символов

Тип	Размер в байтах	Диапазон значений
Variant	Для переменной любого типа	Для переменной любого типа

Переменные и константы типа **Boolean** хранятся как 16-битные числа, но могут принимать только значения **True** или **False**. Используйте эти ключевые слова, чтобы присвоить одно из двух состояний переменной типа **Boolean**. Значение по умолчанию для переменных типа **Boolean** — **False**. Когда числовые типы данных приводятся к типу **Boolean**, 0 становится **False**, а все остальные значения — **True**. И наоборот, когда значение с типом **Boolean** приводится к числовому типу, **False** становится 0, а **True** — 1.

4.6.1.2. Переменные

В языке VBA *переменная* (variable) используется для временного хранения данного в оперативной памяти. Переменная должна быть *определена* прежде, чем ее можно использовать. Определение переменной производится при помощи операторов **Dim**, **Private**, **Static** или **Public**, которые также определяют и *область видимости* (область действия) переменной. Например, следующий оператор объявляет целую переменную (тип **Integer**):

```
Dim N As Integer
```

Если тип при определении переменной опущен, то по умолчанию переменная получает тип **Variant**. Например, следующий оператор определяет переменную x типа **Variant**:

```
Dim x
```

Тип переменной или константы можно также определить непосредственно при ее первом вхождении в текст программы с помощью *специальных символов в конце идентификаторов*: % — для **Integer**, & — для **Long**, ! — для **Single**, # — для **Double**, @ — для **Currency**, \$ — для **String**. Например, идентификатор Primer\$ является идентификатором переменной типа **String**. Однако подобная практика не соответствует хорошему стилю программирования. Приведем еще несколько примеров:

```
IntegerVar% = 123           ' Переменная типа Integer
DoubleVar# = 123.45        ' Переменная типа Double
Const Integerconst% = 123 ' Константа типа Integer
```

Замечание

Если специальный символ в конце идентификатора отсутствует и переменная не определена явным образом, то такой идентификатор соответствует переменной с типом **Variant**.

4.6.1.3. Задание времени жизни и области видимости переменных

Время жизни и область видимости определяют соответственно место использования переменной в приложении, а также время ее существования после создания переменной (время жизни).

Область видимости переменной определяет часть кода, которая "знает" о существовании данной переменной. Так, в частности, при определении переменной в процедуре или функции получить или изменить ее значение можно только из кода этой процедуры или функции. Существуют три типа области видимости переменной.

- ❑ Переменные уровня процедуры или функции распознаются только в процедуре или функции, в которой они описаны. Они определяются в теле процедуры или функции при помощи операторов **Dim**, **Private** или **Static** (по умолчанию **Private**). Такие переменные называются *локальными*. При выходе из процедуры такие переменные становятся недоступными. При выходе из процедуры или функции, в которой при помощи операторов **Dim** или **Private** была определена локальная переменная, такая переменная "уничтожается", т. е. освобождается память, занятая такой переменной. Сказанное не относится к локальным переменным, определенным с помощью служебного слова **Static** — их время жизни максимально.
- ❑ Переменные уровня *контейнера* используются только в контейнере (форме, модуле, классе), в котором они описаны, но не в других контейнерах данного проекта. Контейнерные переменные определяются обязательно при помощи служебного слова **Private** в области описания контейнера, т. е. перед определением процедур и функций.
- ❑ Переменные уровня модуля, описанные при помощи оператора **Public**, являются доступными внутри данного проекта, а также других проектов, которые ссылаются на данный проект. Такие переменные называются *глобальными (открытыми)*. Время жизни открытой переменной совпадает со временем работы программы. *Глобальные переменные могут использоваться только в модулях.*

Еще раз отметим, что закрытая (**Private**) переменная сохраняет свое значение, только, пока активна процедура, функция или контейнер, в которых эта переменная определена. При их завершении значение переменной теряется и при повторном запуске переменную надо заново разместить в памяти и инициализировать. Переменные, описанные с использованием оператора **Static** или **Public**, сохраняют свое значение в течение всего времени работы программы.

Примечание

Синтаксис определения переменных с помощью операторов **Static**, **Private** и **Public** аналогичен синтаксису определения переменных с помощью оператора **Dim** (вместо служебного слова **Dim** можно использовать одно из служебных слов **Static**, **Private** или **Public**).

4.6.1.4. Опция явного определения переменных

Для обязательного определения всех переменных в начале модуля, в так называемой области модуля **General Declarations**, надо поместить директиву **Option Explicit**.

Если данная опция включена, то нужно явно определять все переменные, если же выключена, то всем необъявленным переменным присваивается тип по умолчанию.

Пример:

Option explicit

```
Dim MyVar          ' Определение переменной
MyInt = 10         ' Ошибка: используется неопределенная переменная
MyVar = 10         ' Ошибка отсутствует
```

4.6.1.5. Соглашения о записи имен

Язык VBA является регистрово-независимым языком, т. е. не имеет значение, какими буквами написаны *служебные* слова — прописными или строчными. Более того, встроенный редактор языка VBA автоматическая заменяет регистр, как в служебных словах, так и в определенных явно именах переменных и функций. При присвоении имен подпрограммам, константам, переменным и аргументам необходимо учитывать следующие правила.

- ❑ В языке VBA не различаются строчные и прописные буквы, поэтому, например, идентификатор MyIdentifier и myidentifier являются эквивалентными и указывают на одну область памяти. По умолчанию в языке VBA все одинаковые идентификаторы автоматически приводятся к единообразному виду, соответственно его первому вхождению в текст программного кода.
- ❑ Имена должны начинаться с буквы.
- ❑ Идентификатор (имя) не может содержать пробел, точку, восклицательный знак или специальные символы @, &, \$, # (однако, в конце идентификаторов переменных специальные символы использовать можно).
- ❑ Имена не должны содержать более 255 символов.
- ❑ Не рекомендуется использовать имена, совпадающие с названиями стандартных синтаксических конструкций — служебными словами, функциями, объектами, методами и т. д. Совпадение названий может привести к замещению соответствующего стандартного элемента языка. При возникновении такой ситуации для вызова встроенных функций, операторов или методов необходимо явно указывать связанную с ними библиотеку типов. Например, если была объявлена переменная с именем Left, то функция Left должна вызываться как VBA.Left.
- ❑ Имена, определенные в одной области видимости, должны быть уникальны. Это означает, например, что в одной процедуре нельзя объявить две переменные с одинаковым именем. Однако имя переменной, определенной как глобальная, вполне может совпадать с именем переменной, описанной локально. В этом случае соответствующая глобальная переменная замещается локальной переменной. Локальные переменные, описанные в разных подпрограммах, независимы, поэтому их имена могут совпадать.

Кроме этого, при создании идентификатора рекомендуется использовать *формальные префиксы*, указывающие на тип переменной (табл. 4.4). Использование префиксов создает единообразие в обозначении переменных.

Таблица 4.4. Префиксы переменных, указывающие на их тип

Тип данных	Префикс
Boolean	Bln
Byte	Byt
Currency	Cur
Date	Dtm
Double	Dbl
Integer	Int
Long	Lng
Object	Obj
Single	Sng
String	Str
Пользовательский тип данных	Udt
Variant	Vnt

4.6.1.6. Определение переменной

Определение переменной до ее использования в тексте программы в языке VBA не является обязательным. Если при определении переменной не был явно указан тип переменной или она вообще не была описана, то ей автоматически присваивается тип **Variant**:

```

VarVal = "10"                ' VarVal имеет тип String
                              ' и значение "10"
Dim VarValue As Variant
VarValue = "10"              ' VarValue имеет тип String
                              ' и значение "10"
VarValue = VarValue + 10     ' VarValue имеет тип Integer
                              ' и значение 20
VarValue = VarValue & " штук" ' VarValue имеет тип String
                              ' и значение "20 штук"

```

4.6.1.7. Служебное слово Null

Переменные типа **Variant** могут иметь *особое значение Null*, которое означает, что переменные отсутствуют, неизвестны или неприменимы. Например, по умолчанию данные в полях таблицы базы данных имеют тип **Variant**. Поэтому, если оставить поле пустым, ему будет присвоено значение **Null**. Функция `IsNull` проверяет, является ли указанное значение **Null**:

```

Dim VntVar, BlnVar As Boolean
VntVar = 3
BlnVar = IsNull( VntVar )    ' VntVar примет значение False

```

4.6.1.8. Комментарии

Комментарии в языке VBA пишутся за символом ' вплоть до конца строки. Можно также использовать для этих целей служебное слово **Rem** в начале строки или после оператора. Пример использования комментариев:

```
Dim Inta As Integer
' *****
' * Inta - целая переменная *
' *****
Dim Strb As String: Rem Комментарий д. б. после двоеточия
Rem - это тоже строка комментария
```

4.6.1.9. Перенос строки кода

В соответствии с синтаксисом языка VBA оператор должен располагаться в одной строке. Однако это неудобное ограничение можно обойти. Для этого достаточно использовать комбинацию символов " _" (пробел + символ подчеркивания) в конце строки и последующая строка будет восприниматься как продолжение предыдущей:

```
MsgBox "Имя состоит только" & vbCr _
      & " из букв латинского алфавита"
' MsgBox выводит окно с заданным сообщением
' vbCr осуществляет переход на новую строку
' & является операцией сложения строк
```

4.6.1.10. Строки и строковые операции

Строка представляет собой последовательность символов, которая должна быть окружена кавычками:

```
Dim StrS As String
StrS = "Это строка"
```

В языке VBA имеется две строковые операции: *присваивание* и *конкатенация*. Конкатенация обозначается символом & или символом +:

```
Dim StrS As String
StrS = "Visual Basic " & "for " + "Applications"
```

Служебное слово Empty возвращает ссылку на пустую строку. Того же эффекта можно добиться, поставив рядом две кавычки (""). Например:

```
If Empty = "" Then MsgBox "Равносильны"
```

4.6.1.11. Даты

Тип Date подразумевает как время, так и дату. Отображаются даты из интервала от 1 января 100 года до 31 декабря 9999 года, а время из интервала от 0:00:00 до 23:59:59. Значения дат могут быть представлены в любом распознаваемом формате и должны быть окаймлены знаками "#". Например, следующие два оператора присваивают переменным DtmD1 и DtmD2 одно и то же значение — 31 января 2003 года:

```
Dim DtmD1 As Date, DtmD2 As Date
DtmD1 = #1/31/2003#
DtmD2 = #31 Jan 2003#
```

4.6.1.12. Статические массивы

Массив, как и скалярную переменную, надо определять с помощью операторов **Dim**, **Static**, **Private** и **Public**, которые также задают область видимости и время действия массива. В массиве допускается описание до 60 размерностей. При определении размерности надо указывать верхнюю и нижнюю границы. Если нижний индекс не задан явно, нижняя граница массива определяется директивой **Option Base**. Если данная директива отсутствует, нижняя граница массива равняется нулю:

```
Dim IntA( 11 ) As Integer
```

IntA(0) — первый элемент массива, IntA(11) — последний, 0 — базовый индекс, всего 12 элементов.

Пример определения матрицы из 2 строк и 3 столбцов:

```
Dim SngB( 1, 2 ) As Single
```

Пример изменения нижней границы массива — массив из 11 элементов:

```
Option Base 1
```

```
Dim IntA( 11 ) As Integer
```

При определении массива можно использовать служебное слово **To**:

```
Dim IntA( 1 To 12 ) As Integer
```

Элементы массива можно инициализировать либо последовательностью операторов

```
Dim SngB( 1, 1 ) As Single
```

```
' Два оператора в одной строке
```

```
SngB( 0, 0 ) = 2 : SngB( 0, 1 ) = 4
```

```
SngB( 1, 0 ) = 1 : SngB( 1, 1 ) = 6
```

либо оператором цикла

```
Dim IntM( 1 To 9, 1 To 9 ) As Integer, Inti As Integer, _  
      Intj As Integer
```

```
' Циклы For-Next, переменные цикла Inti и Intj
```

```
' Шаг по умолчанию равен 1
```

```
For Inti = 1 To 9
```

```
    For Intj = 1 To 9
```

```
        IntM( Inti, Intj ) = Inti * Intj
```

```
    Next
```

```
Next
```

Удобным способом определения одномерных массивов является функция **Array**, преобразующая список элементов, разделенных запятыми, в вектор из этих значений и присваивающая их переменной типа **VARIANT**. Допустима инициализация как одномерного массива, так и многомерного, за счет применения вложенных конструкций с функциями **Array**.

Пример инициализации одномерного массива с использованием функции **Array**:

```
Dim VntNum As Variant, DblS As Double
```

```
VntNum = Array( 10, 20 )
```

```
DblS = VntNum( 0 ) + VntNum( 1 )
```

```
MsgBox DblS
```

Пример инициализации двумерного массива с использованием вложенных конструкций с функциями `Array`:

```
Dim VntCityCountry As Variant
VntCityCountry = Array( _
    Array( "Санкт-Петербург", "Россия" ), _
    Array( "Кейптаун", "ЮАР" ) )
' Отобразится "Санкт-Петербург"
MsgBox VntCityCountry( 0 ) ( 0 )
MsgBox VntCityCountry( 0 ) ( 1 ) ' Отобразится "Россия"
```

4.6.1.13. Динамические массивы

Иногда в процессе выполнения программы требуется изменять размер массива. В этом случае первоначально массив определяют как динамический. Для этого при определении массива не надо указывать размерность:

```
Dim SngR( ) As Single
```

Затем в программе следует вычислить необходимый размер массива и изменить размер динамического массива с помощью *оператора **ReDim***.

В следующем примере сначала объявляется динамический массив, а затем устанавливаются границы его индекса:

```
Dim DblR( ) As Double
ReDim DblR( 1 To 10 )
```

Допустимо повторное использование оператора **ReDim** для изменения числа элементов и размерностей массива. В следующем примере создается массив с результатами бросания монеты. Монета бросается до тех пор, пока три раза не выпадет орел. Размерность динамического массива после каждого броска корректируется с сохранением в нем ранее записанных результатов бросания монеты за счет использования *служебного слова **Preserve***.

```
Dim VntAttempt( ), Inti As Integer, IntScore As Integer, _
    IntCoin As Integer
Inti = 0: IntScore = 0
' Цикл Do, выполняется, пока не выполнится условие Until
Do
    Inti = Inti + 1
    IntCoin = Int( 2 * Rnd( ) )
    ' Функция Int( ) возвращает целую часть числа
    ' Функция Rnd( ) возвращает случайное число из
    ' интервала от 0 до 1
    ' 0 - решка, 1 - орел
    If IntCoin = 1 Then IntScore = IntScore + 1
    ReDim Preserve VntAttempt( Inti )
    VntAttempt( Inti ) = Inti
Loop Until IntScore = 3
```

4.6.1.14. Определение границ индексов массива

Функции *LBound*, *UBound* возвращают минимальное и максимальное допустимые значения указанного индекса массива. Например, в следующем коде отобразится 100 и 5:

```
Dim VntA( 1 To 100, 0 To 5 )
MsgBox UBound( VntA, 1 ) & vbCrLf & UBound( VntA, 2 )
' vbCrLf - оператор новой линии
```

Следующие операторы позволяют перебирать элементы массива без указания его размерности:

```
Dim VntD As Variant, Inti As Integer
VntD = Array( "Пн", "Вт", "Ср", "Чт", "Пт" )
For Inti = LBound( VntD ) To UBound( VntD )
    MsgBox VntD( Inti )
Next
```

4.6.1.15. Константы

Константы в отличие от переменных, *не могут изменять свои значения*. Константы могут записываться либо в виде *литералов*, либо определяться как *именованные константы*. Именованные константы определяются при помощи служебного слова **Const**, перед которым, не обязательно, может находиться определяющее область видимости служебное слово **Public** или **Private** (при его отсутствии предполагается по умолчанию **Private**). Как и переменные, именованные константы делятся на три группы — локальные, контейнерные и глобальные. Если именованная константа определяется внутри процедуры или функции (**Private**), то она является локальной и доступна только внутри процедуры или функции. Контейнерные именованные константы доступны только внутри контейнера (формы, модуля или класса), в котором они определены. Глобальные именованные константы могут определяться только в модуле с использованием служебного слова **Public**. Они доступны во всей программе. Приведем примеры констант-литералов и именованных констант:

```
' Именованные константы для типа Byte
' *****
Public Const BytB1 As Byte = 255
Private Const BytB2 As Byte = 0      ' или эквивалентно
Const BytB2 As Byte = 0
Const BytB2 As Byte = 121
' Литералы и именованные константы для типа Currency
' *****
1000000.0000@          ' Литерал
Public Const CurC1 As Currency = 1000000.0000@
Private Const CurC2 As Currency = 1000000.0000@
' или эквивалентно
Const CurC3 As Currency = 1000000.0000@
' Литералы и именованные константы для типа Date
' *****
#January 1, 1993#      ' Литерал
Public Const DtmD1 = #January 1, 1993#
```

```

Private Const DtmD2 = #January 1, 1993#
' или эквивалентно
Const DtmD3 = #January 1, 1993#
' Литералы и именованные константы для типа Double
' *****
1.5 ' Литерал
Public Const DblD1 As Double = 1.5
Private Const DblD2 As Double = 1.5
' или эквивалентно
Const DblD3 As Double = 1.5
' Литералы и именованные константы для типа Integer
' *****
1 ' Литерал
Public Const IntI1 As Integer = 2
Private Const IntI2 As Integer = 3
' или эквивалентно
Const IntI3 As Integer = 3
' Литералы и именованные константы для типа Long
' *****
1& ' Литерал
Public Const LngL1 As Long = 2&
Private Const LngL2 As Long = 3&
' или эквивалентно
Const LngL3 As Long = 3&
' Литералы и именованные константы для типа Single
' *****
17.4! ' Литерал
Public Const SngS1 As Single = 17.4!
Private Const SngS2 As Single = 17.4!
' или эквивалентно
Const SngS2 As Single = 17.4!
' Литералы и именованные константы для типа String
' *****
"Hello" ' Литерал
Public Const StrS1 As String = "Hello"
Private Const StrS2 As String = "Hello"
' или эквивалентно
Const StrS2 As String = "Hello"

```

4.6.1.16. Перечисляемый тип

Перечисляемый тип предоставляет удобный способ работы с целочисленными константами и позволяет ассоциировать их значения с именами. Этот тип определяется при помощи *служебного слова* **Enum**, перед которым может идти модификатор доступа **Public** или **Private**. В следующем далее примере задается перечисляемый тип для идентификации стороны монеты:

```

Enum Coin
    Head = 1
    Tail = -1
End Enum
' Подпрограмма Attempt( )

```

```

Sub Attempt( )
  Dim r As Integer
  ' Оператор Randomize используется для указания выборки
  ' псевдослучайных чисел
  Randomize
  r = 2 * Int( 2 * Rnd( ) ) - 1
  ' Оператор выбора
  Select Case r
    Case Head
      MsgBox "Орел!"
    Case Tail
      MsgBox "Решка"
  End Select
End Sub

```

Если перечисляемым константам целого типа (в данном случае Head и Tail) явно не заданы значения, то по умолчанию они полагаются равными 0, 1, ...

4.6.1.17. Тип данных, определяемых пользователем (структурный тип)

Наряду с массивами, представляющими нумерованный набор элементов одинакового типа, существует еще один способ создания *структурного типа* — определенный пользователем тип или, в привычной терминологии для программистов — *запись (структура)*. Запись — это совокупность нескольких элементов, каждый из которых может иметь свой тип. Элемент записи называется полем. Запись является частным случаем класса, в котором не определены методы. Данный тип определяется при помощи *служебного слова Type*, перед которым может идти модификатор доступа **Public** или **Private**. Оператор **Type** используется *только на уровне модуля*. Появлению в модуле класса оператора **Type** должно предшествовать служебное слово **Private**. В следующем далее примере оператор **Type** используется для определения типа данных MyType, инкапсулирующего в себе информацию о персонажах сказок. У этой записи имеются три поля. Первое поле имеет тип **Integer** и содержит идентификационный номер героя, второе поле имеет тип **String** и специфицирует его имя. Третье поле имеет тип **Variant** и может содержать любой тип данных, и даже динамический массив.

```

' Объявление структурного типа
Private Type MyType
  ID As Integer
  Name As String
  Info As Variant
End Type
' Использование структурного типа
Sub TestMyType( )
  ' Массив структур
  Dim a( 1 ) As MyType
  ' Динамический массив строк для поля Variant
  Dim txt( ) As String
  ' Пересоздание и инициализация массива строк
  ReDim txt( 1 )

```

```

txt( 0 ) = " из"
txt( 1 ) = " страны чудес"
' Инициализация первого элемента массива структур
a( 0 ).ID = 2
a( 0 ).Name = " Алиса"
a( 0 ).Info = txt
ReDim txt( 2 )
txt( 0 ) = " Пух"
txt( 1 ) = " - наш любимый "
txt( 2 ) = "персонаж"
a( 1 ).ID = 1
a( 1 ).Name = " Винни"
a( 1 ).Info = txt
' Вывод в окно Immediate
Debug.Print "    a0    "
Debug.Print a( 0 ).ID & a( 0 ).Name _
            & a( 0 ).Info( 0 ) & a( 0 ).Info( 1 )
Debug.Print "    a1    "
Debug.Print a( 1 ).ID & a( 1 ).Name _
            & a( 1 ).Info( 0 ) & a( 1 ).Info( 1 ) _
            & a( 1 ).Info( 2 )
End Sub

```

4.6.1.18. Приоритеты операций

Выражения в языке VBA состоят из операндов (константы, переменные, указатели функций и подвыражения) и операций. При вычислении значения выражения порядок выполнения операций определяется их приоритетами. Приоритет операций позволяет установить последовательность выполнения вычислений. В языке VBA существует несколько правил, строго регламентирующих порядок вычисления значения выражения.

- Операции, расположенные внутри круглых скобок, всегда выполняются раньше, чем операции вне скобок.
- При проведении вычислений в смешанных выражениях, содержащих операции разных типов, сначала выполняются арифметические операции, затем операции сравнения и логические операции.
- Арифметические операции выполняются в следующем порядке по убыванию приоритетов: возведение в степень **^**, изменение знака (унарный минус **-**), равнозначные по приоритету умножение и деление ***** и **/**, деление по модулю **Mod**, равнозначные по приоритету сложение и вычитание (бинарные операции плюс **+** и минус **-**), сложение строк **&**. Операции одинакового приоритета выполняются в порядке их расположения в выражении слева направо.
- Операции сравнения имеют равный приоритет и выполняются в порядке их расположения в выражении слева направо.
- Логические операции выполняются в следующем порядке по убыванию их приоритетов: **Not**, **And**, **Or**, **Xor**, **Eqv**, **Imp**.

4.6.1.19. Математические операции

В языке VBA поддерживается стандартный набор *математических операций*. Следующий далее код демонстрирует их применение:

```
Dim x As Double, y As Double, z As Double
x = 1 : y = 3 ' Инициализация переменных в одной строке
z = x + y ' Сложение
z = x - y ' Вычитание
z = x * y ' Умножение
z = x / y ' Деление
' Целочисленное деление (результат – целое)
z = x \ y
z = x Mod y ' Остаток от деления по модулю
z = x ^ y ' Возведение в степень
```

4.6.1.20. Операции отношения

В языке VBA используется стандартный набор *операций отношения*:

```
Dim x As Double, y As Double, b As Boolean
x = 1 : y = 3
b = ( x < y ) ' Меньше
b = ( x <= y ) ' Меньше или равно
b = ( x > y ) ' Больше
b = ( x >= y ) ' Больше или равно
b = ( x <> y ) ' Не равно
b = ( x = y ) ' Равно
```

4.6.1.21. Логические операции

Логические операции языка VBA перечислены в табл. 4.5.

Таблица 4.5. Логические операции языка VBA

Операции	Описание
exp1 And exp2	Логическое умножение (результат True при exp1=exp2= True)
exp1 Or exp2	Логическое сложение (результат True при exp1 и/или exp2 True)
exp1 Xor exp2	Исключающее ИЛИ (результат True тогда и только тогда, когда только один операнд имеет значение True)
Not exp	Логическое отрицание (результат при exp= True и наоборот)
exp1 Imp exp2	Логическая импликация (принимает значение False лишь в случае, когда exp1= True , а exp2= False)
exp1 Equ exp2	Логическая эквивалентность (результат True , когда операнды равны друг другу)
exp1 Like exp2	Операция сравнения объектов типа String (результат True , если строки равны друг другу)

Действие операции сравнения **Like** зависит от директивы **Option Compare**, которая располагается в области описания модуля. По умолчанию для каждого модуля

считается установленной инструкция **Option Compare Binary**, при которой различаются строчные и прописные буквы, т. е. следующий оператор вернет **False**:

```
Debug.Print "AA" Like "aa"
```

Если же в области описания модуля указан оператор **Option Compare Text**, то при сравнении строчные и прописные буквы не различаются, и тот же самый оператор на этот раз вернет **True**.

В следующем далее примере в поле ввода диалогового окна пользователь вводит свое имя латинскими буквами. При нажатии кнопки **ОК** программа анализирует информацию и отображает соответствующее сообщение:

- если пользователь забыл ввести имя, то его об этом информируют;
- если во введенном имени присутствуют знаки, отличные от букв латинского алфавита, пользователя об этом информируют;
- если имя состоит только из букв латинского алфавита, то с пользователем здороваются.

Option Compare Text

```
Sub DemoCompare( )
    Dim StrName As String, IntLng As Integer, _
        IntI As Integer
    StrName = InputBox( "Введите имя" )
    ' InputBox( ) выводит окно с полем ввода. Возвращает
    ' введенное выражение
    StrName = Trim( StrName )
    ' Trim( ) урезает лидирующие и последние пробелы в
    ' строке
    IntLng = Len( StrName )
    ' Len( ) возвращает длину строки
    If IntLng = 0 Then
        MsgBox "Забыли ввести имя"
        Exit Sub
    Else
        For IntI = 1 To IntLng
            If Not Mid( StrName, IntI, 1 ) _
                Like "[A-Z]" Then
                ' Функция Mid( ) возвращает подстроку строки,
                ' содержащую, начиная со специфицированной
                ' позиции, указанное число символов
                ' "[A-Z]" - строка, содержащая все символы,
                ' находящиеся между A и Z
                MsgBox "Имя должно состоять только" _
                    & vbCrLf & _
                    "из букв латинского алфавита"
                Exit Sub
            End If
        Next IntI
    End If
    MsgBox "Привет, " & StrName
End Sub
```

4.6.1.22. Операции присваивания

Оператор присваивания присваивает значение выражения переменной, константе или свойству объекта (свойство — элемент ООП, позволяющий получать и устанавливать значения параметров текущего состояния объекта). Оператор присваивания всегда включает знак равенства =. Например, в результате, действия следующей пары операторов

```
x = 2
x = x + 2
```

переменной x будет присвоено значение 4.

Для присваивания переменной ссылки на объект в операторе присваивания применяется служебное слово **Set**. В общем случае оператор **Set** имеет следующий синтаксис:

```
Set varname = {[ New ] expression | Nothing}
```

Здесь служебное слово **New** используется при создании нового экземпляра класса, а служебное слово **Nothing** позволяет освободить все системные ресурсы и ресурсы памяти, выделенные для объекта, на который имелась ссылка (проще говоря, объект удаляется из памяти). Пример:

```
' Определение объекта типа Drives
Dim objFSO As FileSystemObject
Dim objDrives As Drives
Set objFSO = New FileSystemObject
Set objDrives = objFSO.Drives
```

4.6.1.23. Оператор With

Оператор With избавляет программиста от утомительной обязанности использовать большое количество повторений имени одного и того же объекта при работе с его свойствами и методами. Кроме того, он структурирует код, делая его более прозрачным:

```
With frmFirst
    .Font.Italic = True
    .PictureTiling = True
End With
```

Это эквивалентно следующей записи:

```
frmFirst.Font.Italic = True
frmFirst.PictureTiling = True
```

4.6.1.24. Операторы управления

В языке VBA имеются несколько операторов управления ходом выполнения программы. Функционально они делятся на две группы:

- операторы перехода и выбора (**Goto**, **If** и **Select**);
- операторы повтора (**For Next**, **For Each**, **Do-Loop** и **While**).

Оператор условного перехода If задает выполнение определенных групп операторов в зависимости от значения выражения. Например, если скидка (скажем, 5%)

применяется только к суммам больше 1000, то в языке VBA это можно записать следующим образом:

```
If Сумма > 1000 Then Скидка = 0.05 Else Скидка = 0
```

что эквивалентно

```
Скидка = 0
```

```
If Сумма > 1000 Then Скидка = 0.05
```

Рекомендуется использование блочной формы синтаксиса, которая часто упрощает восприятие оператора условного перехода. Приводимый ранее пример со скидкой можно записать в следующей эквивалентной блочной структуре:

```
If Сумма > 1000 Then
```

```
    Скидка = 0.05
```

```
Else
```

```
    Скидка = 0
```

```
End If
```

или

```
Скидка = 0
```

```
If Сумма > 1000 Then
```

```
    Скидка = 0.05
```

```
End If
```

Дерево условий может оказаться гораздо более сложным, чем просто проверка одного условия. В этом случае используется оператор *If Then ElseIf*, который позволяет проверять множественные условия. Следующий пример демонстрирует то, как производится порядок проверки условий. В нем, в зависимости от величины вводимого числа, отображается сообщение о принадлежности числа либо интервалу [0, 1], либо интервалу (1, 2], либо о не принадлежности числа этим двум интервалам.

```
Sub DemoElseIf( )
```

```
    x = InputBox( "Введите число" )
```

```
    ' Преобразование строки в число
```

```
    x = Val( x )
```

```
    If 0 <= x And x <= 1 Then
```

```
        MsgBox "Число из интервала [0, 1]"
```

```
    ElseIf 1 < x And x <= 2 Then
```

```
        MsgBox "Число из интервала (1, 2]"
```

```
    Else
```

```
        MsgBox "Число либо отрицательное, либо больше 2"
```

```
    End If
```

```
End Sub
```

Не применяйте оператор условного перехода, если это возможно :

```
Sub TrueOrFalseSlower( )
```

```
    Dim BlnIsYes As Boolean
```

```
    Dim IntI As Integer
```

```
    IntI = 3
```

```
    If IntI = 5 Then
```

```
        BlnIsYes = True
```

```
    Else
```

```

        BlnIsYes = False
    End If
    MsgBox BlnIsYes
End Sub

```

В этом случае предпочтительнее использовать приводимый далее код, который не только изящнее, но и работает быстрее:

```

Sub TrueOrFalseFaster( )
    Dim BlnIsYes As Boolean
    Dim IntI As Integer
    IntI = 3
    BlnIsYes = ( IntI = 5 )
    MsgBox BlnIsYes
End Sub

```

Оператор выбора Select Case выполняет одну из нескольких групп операторов в зависимости от значения выражения. Оператор выбора очень эффективен, когда надо проверить одну переменную или выражение, принимающие ограниченное количество значений (типы **Byte**, **Integer**, **Long**). В следующем примере, как и ранее, в зависимости от величины введенного числа, отображается сообщение, указывающее на величину числа или диапазон, которому оно принадлежит.

```

Sub DemoSelect( )
    Dim IntX As Integer
    IntX = InputBox( "Введите целое число" )
    Select Case IntX
        Case 1
            MsgBox "Число равно 1"
        Case 2, 3
            MsgBox "Число равно 2 или 3"
        Case 4 To 6
            MsgBox "Число от 4 до 6"
        Case Is >= 7
            MsgBox "Число не менее 7"
        Case Else
            MsgBox " Число менее 1"
    End Select
End Sub

```

Оператор For Next повторяет выполнение группы операторов указанное число раз, а именно, пока переменная цикла изменяется от начального значения до конечного с указанным шагом. Если шаг не указан, то он полагается равным 1. Альтернативный способ выхода из цикла предоставляет оператор **Exit For**. В следующем примере при помощи оператора цикла находится сумма элементов массива.

```

Sub DemoFor1( )
    Dim VntA As Variant
    VntA = Array( 1, 4, 12, 23, 34, 3, 23 )
    VntS = 0
    For Vnti = LBound( VntA ) To UBound( VntA )
        VntS = VntS + VntA( Vnti )
    Next

```

```

    MsgBox VntS
End Sub

```

В следующем коде находится сумма всех четных целых из интервала от 1 до 100.

```

Sub DemoFor2( )
    Dim IntI As Integer, IntS As Integer
    IntS = 0
    For IntI = 2 To 100 Step 2
        IntS = IntS + IntI
    Next
    MsgBox IntS
End Sub

```

Оператор For Each повторяет выполнение группы операторов для каждого элемента массива или семейства. Альтернативный способ выхода из цикла предоставляет оператор **Exit For**. В следующем далее коде оператор **For Each** используется для суммирования элементов массива:

```

Sub DemoForEach( )
    Dim VntA As Variant, DblS As Double
    VntA = Array( 1, 4, 12, 23, 34, 3, 23 )
    DblS = 0
    For Each b In VntA
        DblS = DblS + b
    Next
    MsgBox DblS
End Sub

```

Оператор While выполняет последовательность операторов, пока заданное условие возвращает значение **True**. Оператор повтора **While** в отличие от оператора **For** выполняется не заданное число раз, а пока выполняется условие. В следующем примере бросается игральная кость до тех пор, пока не выпадет шесть очков. При выпадении шести очков игра заканчивается, и отображается сообщение с указанием, на каком броске она закончилась.

```

Sub DemoWhile( )
    Dim IntAttempt As Integer, IntScore As Integer
    Randomize
    IntScore = Int( 6 * Rnd( ) ) + 1
    IntAttempt = 1
    While IntScore < 6
        IntAttempt = IntAttempt + 1
        IntScore = Int( 6 * Rnd( ) ) + 1
    Wend ' Служебное слово окончания цикла While
    MsgBox "Победили на попытке: " & IntAttempt
End Sub

```

Оператор Do повторяет выполнение набора операторов, пока условие имеет значение **True** (случай **While**) или пока оно не примет значение **True** (случай **Until**):

```

Do [{While | Until} condition]
    [statements]
Exit Do
    [statements]

```

```

Loop
или
Do
    [statements]
    [Exit Do]
    [statements]
Loop [{While | Until} condition]

```

В любом месте управляющей структуры **Do** может быть размешено любое число операторов **Exit Do**, обеспечивающих альтернативные возможности выхода из цикла **Do**. Примером использования оператора цикла **Do Until** может быть следующий далее код, который обеспечивает повторение цикла до тех пор, пока в поле ввода диалогового окна не будет введен пароль (в данном случае Admin):

```

Sub DemoPassword( )
    Dim StrPs As String
    Do
        StrPs = InputBox( "Введите пароль" )
    Loop Until StrPs = "Admin"
End Sub

```

или эквивалентно

```

Sub DemoPassword( )
    Dim StrPs As String
    Do While StrPs <> "Admin"
        StrPs = InputBox( "Введите пароль" )
    Loop
End Sub

```

*Оператор безусловного перехода **Goto*** задает переход на указанную строку внутри подпрограммы или функции. Для использования оператора безусловного перехода надо какой-то строке присвоить метку. Метка должна начинаться с буквы и заканчиваться двоеточием. В качестве примера использования оператора безусловного перехода рассмотрим далее игру, в которой игроку даны десять попыток броска игральной кости. В случае если при какой-то из этих попыток выпадает шесть очков, игрок выигрывает, и игра заканчивается.

```

Sub DemoGoTo( )
    Dim IntI As Integer, IntScore As Integer
    Randomize
    For IntI = 1 To 10
        IntScore = Int( 6 * Rnd( ) ) + 1
        If IntScore = 6 Then Goto lblMessage
    Next
    Goto lblEnd
    lblMessage: MsgBox "Выиграли при броске " & IntI
    lblEnd:
End Sub

```

4.6.1.25. Основные программы и функции языка

Язык VBA содержит большое количество *стандартных подпрограмм и функций*, которые существенно расширяют возможности языка. Далее рассматриваются основные подпрограммы и функции, сгруппированные по их назначению.

4.6.1.25.1. Математические функции

В языке VBA имеется представительная группа *математических функций* (табл. 4.6).

Таблица 4.6. Математические функции языка VBA

Функция	Описание
Abs	Модуль (абсолютная величина) аргумента функции
Atn	Арктангенс аргумента функции, выраженного в радианах
Cos	Косинус аргумента функции, выраженного в радианах
Exp	Экспонента, т. е. результат возведения основания натурального логарифма в степень, указанную аргументом функции
Log	Натуральный логарифм аргумента функции
Rnd	Функция Rnd(number) возвращает очередное псевдослучайное число из интервала [0, 1]. Если значение параметра number меньше нуля, то возвращается число, зависящее от number. Если значение параметра number больше нуля или этот параметр опущен, то возвращает следующее случайное число в последовательности
Sgn	Знак аргумента функции (1 — аргумент положительный, 0 — аргумент нулевой, -1 — аргумент отрицательный)
Sin	Синус аргумента функции, выраженного в радианах
Sqr	Квадратный корень из аргумента функции
Tan	Тангенс аргумента функции, выраженного в радианах
Fix, Int	Функции отбрасывают дробную часть числа и возвращают целое значение. Различие между ними состоит в том, что для отрицательного значения аргумента функция Int возвращает отрицательное ближайшее целое число, меньшее либо равное аргументу, а Fix — ближайшее отрицательное целое число, большее либо равное аргументу

Функция Rnd генерирует очередное число из последовательности псевдослучайных чисел. У таких последовательностей, если совпадают первые члены, то совпадают и все последующие. Перед вызовом функции Rnd следует применять *оператор Randomize*, который определяет первый член этой последовательности. Если этот оператор используется без параметра, то первый член последовательности привязан к текущему системному времени, что делает число, возвращаемое функцией Rnd, "более" случайным. В языке VBA нет функции, возвращающей число π , но для его нахождения можно использовать функцию Atn:

```
Dim Pi As Double
Pi = 4.0 * Atn( 1 )
```

Конечно, можно задать n и явным образом, указав достаточное число значащих цифр, но этот подход менее удобен.

4.6.1.25.2. Функции проверки типов

Функция проверки типа проверяет, имеет ли ее аргумент заданный тип (табл. 4.7).

Таблица 4.7. Функции проверки типов языка VBA

Функция	Проверяемый тип
IsArray	Возвращает True , если аргумент функции является массивом
IsDate	Возвращает True , если аргумент функции является датой
IsEmpty	Возвращает True , если аргумент функции был явно объявлен
IsError	Возвращает True , если аргумент функции является кодом ошибки
IsNull	Возвращает True , если аргумент функции имеет значение Null
IsNumeric	Возвращает True , если аргумент функции является значением переменной
IsObject	Возвращает True , если аргумент функции является объектом

4.6.1.25.3. Функции преобразования типов

Преобразование строки в число. Функция Val возвращает подходящее число, заданное в аргументе функции в виде строки. При наличии другого десятичного разделителя целой и дробной частей (например, запятой) будет преобразована только часть строки-аргумента, предшествующая недопустимому разделителю. Приведем далее примеры преобразования строк в числа:

```

Sub DemoStrToNum( )
    Dim StrS As String
    StrS = "23432"
    Debug.Print Val( StrS )           ' Отобразится 23432
    StrS = "123-45-45"
    Debug.Print Val( StrS )           ' Отобразится 123
    StrS = "#12/15/1999#"
    Debug.Print Val( StrS )           ' Отобразится 15,12
    StrS = "198,005"
    Debug.Print Val( StrS )           ' Отобразится 198
    StrS = "198.005"
    Debug.Print Val( StrS )           ' Отобразится 198,005
End Sub

```

Преобразование числа в строку. Функция CStr возвращает значение типа **String**, являющееся строковым представлением числового аргумента функции. Например, следующий далее код проверяет, является ли 0 одной из цифр сгенерированного целого случайного числа из интервала от 1 до 1000. Для этого число преобразуется в строку, а затем с помощью функции InStr проверяется, является ли указанная строка подстрокой данной.

```

Sub DemoNumToStr( )
    Dim IntI As Integer
    IntI = 1000 * Rnd( ) + 1

```

```

If InStr( CStr( IntI ), "0" ) > 0 Then
    MsgBox "0 встретился"
Else
    MsgBox "0 не встретился"
End If
End Sub

```

Кроме функций Val и CStr, в языке VBA имеется целый ряд *функций преобразования типов* (табл. 4.8).

Таблица 4.8. Функции преобразования типов языка VBA

Функция	Тип, в который преобразуется аргумент функции (возвращаемое значение)
CBool	Boolean
CByte	Byte
CCur	Currency
CDate	Date
CDBl	Double
CDec	Decimal
CInt	Integer
CLng	Long
CSng	Single
CVar	Variant
CStr	String

4.6.1.25.4. Форматирование числового значения функцией Format

Чтобы представить числовое значение, заданное выражением, как дату, время, денежное значение или в специальном формате, следует использовать *функцию Format*, которая возвращает значение типа **Variant (String)**, содержащее выражение, отформатированное в соответствии с описанием формата:

```

Format( Expression[, NameUserFmt[, FirstDayOfWeek
    [, FirstWeekOfYear]]) ],

```

где Expression — любое допустимое выражение; NameUserFmt — любое допустимое именованное или определяемое пользователем выражение формата; FirstDayOfWeek — константа, определяющая первый день недели и FirstWeekOfYear — константа, определяющая первую неделю года. Примером именованного формата является Fixed — формат для вывода вещественного числа с двумя значащими цифрами после десятичной точки. *Именованные числовые форматы и форматы даты и времени* приведены далее в табл. 4.9 и 4.10).

Таблица 4.9. Именованные числовые форматы

Формат	Описание
General Number	Число без разделителя тысяч
Currency	Использует установки страны из панели управления. Отображает две цифры справа от десятичной точки
Fixed	Отображает, по крайней мере, одну цифру слева и две справа от десятичной точки
Standard	Отображает, по крайней мере, одну цифру слева и две справа от десятичной точки и выводит разделитель тысяч
Percent	Отображает число в виде процентов и выводит две цифры справа от десятичной точки (числовое значение умножается на 100)
Scientific	Использует формат с плавающей десятичной точкой (мантисса-порядок). В мантиссе выводится одна значащая цифра слева от десятичной точки и две цифры справа. Порядок начинается с буквы "E", за которой следует знак порядка и порядок.
Yes/No	Отображает No (Нет), если число равно 0 и Yes (Да) в противном случае
True/False	Отображает False (Ложь), если число равно 0 и True (Истина) в противном случае
On/Off	Отображает Off (Выкл), если число равно 0 и On (Вкл) в противном случае

Таблица 4.10. Именованные форматы даты и времени

Формат	Описание
General Date	Выводит дату или время. Если нет дробной части, то выводит только дату
Long Date	Выводит дату в соответствии с полным форматом ОС Windows для даты
Medium Date	Выводит дату в соответствии с обычным форматом ОС Windows для даты
Short Date	Выводит дату в соответствии с сокращенным форматом ОС Windows для даты
Long Time	Выводит часы, минуты и секунды
Medium Time	Выводит часы и минуты в 12-часовом формате
Short Time	Выводит часы и минуты в 24-часовом формате

Далее приводится пример использования именованных форматов в функции `Format`:

```

Sub DemoUsgFormatFunction( )
    Dim x As Double
    x = 4654646.544564
    MsgBox Format( x, "General Number" ), , _
        "General Number"
    ' Аргументы , , "General Number" задают заголовок окна
    ' сообщения

```

```

' Будет выведено: 4654646,544564
MsgBox Format( x, "Currency" ), , "Currency"
' Будет выведено: 4 654 646,54p.
MsgBox Format( x, "Fixed" ), , "Fixed"
' Будет выведено: 4654646,54
MsgBox Format( x, "Standard" ), , "Standard"
' Будет выведено: 4 654 646,54
MsgBox Format( x, "Percent" ), , "Percent"
' Будет выведено: 465464654,46%
MsgBox Format( x, "Scientific" ), , "Scientific"
' Будет выведено: 4,65E+06
MsgBox Format( x, "Yes/No" ), , "Yes/No"
' Будет выведено: Да
MsgBox Format( x, "True/False" ), , "True/False"
' Будет выведено: Истина
MsgBox Format( x, "On/Off" ), , "On/Off"
' Будет выведено: Вкл
MsgBox Format( Now, "General Date" ), , "General Date"
' Служебное слово Now используется для получения
' текущей даты и времени
' Будет выведено: 05.01.2007 11:10:51
MsgBox Format( Now, "Long Date" ), , "Long Date"
' Будет выведено: 5 Январь 2007 г.
MsgBox Format( Now, "Medium Date" ), , "Medium Date"
' Будет выведено: 05-январь-07
MsgBox Format( Now, "Short Date" ), , "Short Date"
' Будет выведено: 05.01.2007
MsgBox Format( Now, "Long Time" ), , "Long Time"
' Будет выведено: 11:12:26
MsgBox Format( Now, "Medium Time" ), , "Medium Time"
' Будет выведено: 11:12
MsgBox Format( Now, "Short Time" ), , "Short Time"
' Будет выведено: 11:13

```

End Sub

Наряду с именованными числовыми форматами и форматами даты и времени, в функции `Format` можно применять *пользовательские форматы*, позволяющие настроить вид отображаемых значений по желанию пользователя. При построении пользовательского формата применяются специальные символы, приведенные далее в табл. 4.11.

Таблица 4.11. Символы, используемые при построении пользовательского формата

Символ	Описание
0	Резервирует позицию цифрового разряда. Отображает цифру или ноль. Если у числа, представленного параметром, есть какая-нибудь цифра в той позиции разряда, в которой в строке формата находится ноль, то функция отображает эту цифру параметра, а если нет — в этой позиции отображается ноль

Символ	Описание
#	Резервирует позицию цифрового разряда. Отображает цифру или ничего. Если у числа, представленного параметром, есть какая-нибудь цифра в той позиции разряда, в которой в строке формата находится символ #, то функция отображает эту цифру параметра, а если нет — в этой позиции не отображается ничего. Действие этого символа аналогично действию символа ноль, за исключением того, что лидирующие и завершающие нули не отображаются
.	Резервирует позицию десятичного разделителя. Указание точки в строке формата определяет, сколько разрядов необходимо отображать слева и справа от десятичной точки
%	Задаёт процентное отображение числа
,	Задаёт разделитель разряда сотен от тысяч
:	Задаёт разделитель часов, минут и секунд в категории форматов времени
/	Задаёт разделитель дня, месяца и года в категории форматов даты
E+, E-, e+, e-	Задаёт разделитель мантиссы и порядка в экспоненциальном формате
d, m, y	Резервирует позицию при выводе дня, месяца, года в категории форматов даты
h, m, s	Резервирует позицию при выводе часа, минуты, секунды в категории форматов времени

Далее приведен пример использования пользовательских форматов:

```
Sub DemoUsgFormatFunction( )
  MsgBox Format( 1.2^2, "##.###" )
  ' Будет выведено: 1,44
  MsgBox Format( 1.2^2, "##.000" )
  ' Будет выведено: 1,440
  MsgBox Format( Sin( 1 )*Exp( 5 ), "#.##e+##" )
  ' Будет выведено: 1,25e+2
  MsgBox Format( Now, "hh:mm:ss" )
  ' Будет выведено: 13:17:12
  MsgBox Format( Now, "dd/mm/yyyy" )
  ' Будет выведено: 05.01.2007
End Sub
```

4.6.1.25.5. Форматирование числовых, процентных, денежных значений, значений даты и времени специализированными функциями

Наряду с универсальной функцией `Format`, для форматирования выводимых значений можно использовать специализированные функции `FormatNumber` (форматирование числовых значений), `FormatPercent` (форматирование процентных значений), `FormatCurrency` (форматирование денежных значений), и `FormatDateTime` (форматирование даты и времени).

Для форматирования числовых значений можно использовать специализированную функцию `FormatNumber`, имеющую следующий формат:

```
FormatNumber( Expression[, NumDigitsAfterDecimal
```

```
[, IncludeLeadingDigit[,
UseParensForNegativeNumbers
[, GroupDigits]]] ),
```

где Expression — обязательный параметр, указывающий форматлируемое числовое выражение; NumDigitsAfterDecimal — необязательный параметр, задающий число знаков, отображаемых после десятичной точки (по умолчанию в дробной части выводятся две цифры); IncludeLeadingDigit — необязательный параметр, указывающий, надо ли отображать целую нулевую часть (допустимыми значениями являются константы: vbTrue — отображать, vbFalse — не отображать и vbUseDefault — отображать); UseParensForNegativeNumbers — необязательный параметр, определяющий, надо ли отрицательные числа отображать в скобках (допустимыми значениями являются константы vbTrue — отображать, vbFalse — не отображать и vbUseDefault — не отображать); GroupDigits — необязательный параметр, определяющий, надо ли группировать цифры (допустимыми значениями являются константы vbTrue — группировать, vbFalse — не группировать и vbUseDefault — группировать).

Далее приведен пример использования функции FormatNumber:

```
Sub DemoUsgFormatNumberFunction( )
  MsgBox FormatNumber( Sin( 4 ) )
  ' Будет выведено: -0,76
  MsgBox FormatNumber( Sin( 4 ), 3 )
  ' Будет выведено: -0,757
  MsgBox FormatNumber( Sin( 4 ), 3, vbTrue )
  ' Будет выведено: -0,757
  MsgBox FormatNumber( Sin( 4 ), 3, vbFalse )
  ' Будет выведено: -,757
  MsgBox FormatNumber( Sin( 4 ), 3, vbFalse, vbTrue )
  ' Будет выведено: (,757)
  MsgBox FormatNumber( Sin( 4 ), 3, vbFalse, vbFalse )
  ' Будет выведено: -,757
  MsgBox FormatNumber( 14547, 3, vbFalse, vbFalse, _
    vbTrue )
  ' Будет выведено: 14 547,000
```

End Sub

Для *форматирования процентных значений* можно использовать специализированную функцию FormatPercent, имеющую такой же синтаксис, как и функция FormatNumber. Например, следующая далее функция

```
Sub DemoUsgFormatPercentFunction( )
  x = 0.2342
  MsgBox FormatPercent( x, 2 )
```

End Sub

выведет значение 23.42%, т. е. число, записанное в формате процентов, у которого после десятичной точки отображаются две цифры.

Для *форматирования денежных значений* можно использовать специализированную функцию FormatCurrency, имеющую такой же синтаксис, как и функция FormatNumber. Например, следующая далее функция

```

Sub DemoUsgFormatPercentFunction( )
    MsgBox FormatCurrency( 12312.3453, 2 )
End Sub

```

выведет значение 12 312,35р.

Для *форматирования даты и времени* можно использовать специализированную функцию `FormatDateTime`, имеющую следующий синтаксис:

```
FormatDateTime( Date[, NamedFormat] ),
```

где `Date` — обязательный параметр, задающий форматируемую дату; `NamedFormat` — необязательный параметр, указывающий тип форматирования (допустимыми значениями являются константы `vbGeneralDate`, `vbLongDate`, `vbShortDate`, `vbLongTime`, `vbShortTime`).

Далее приведен пример использования функции `FormatDateTime`:

```

Sub DemoUsgFormatDateTimeFunction( )
    MsgBox FormatDateTime( Now, vbGeneralDate )
    ' Будет выведено: 05.01.2007 15:33:35
    MsgBox FormatDateTime( Now, vbLongDate )
    ' Будет выведено: 5 Январь 2007 г.
    MsgBox FormatDateTime( Now, vbShortDate )
    ' Будет выведено: 05.01.2007
    MsgBox FormatDateTime( Now, vbLongTime )
    ' Будет выведено: 15:34:21
    MsgBox FormatDateTime( Now, vbShortTime )
    ' Будет выведено: 15:34
End Sub

```

End Sub

4.6.1.26. Процедуры и функции

Процедура является самостоятельной частью кода, которая имеет имя и может иметь параметры, выполнять последовательность операторов и изменять значения своих параметров. Процедура имеет следующий синтаксис:

```

[Private|Public|Friend] [Static] Sub name([ arglist ])
    [statements]
    [Exit Sub]
    [statements]

```

End Sub

Здесь **Public** — служебное слово, указывающее на то, что процедура является *открытой* и доступна для всех других процедур во всех модулях, кроме модулей, содержащих оператор **Options Private**; **Private** — служебное слово, указывающее на то, что процедура является *закрытой*, и областью ее видимости является текущий модуль; **Friend** — служебное слово, используемое только в модуле класса и указывающее на то, что процедура является *дружественной*, и областью ее видимости является проект, содержащий модуль класса; **Static** — служебное слово, указывающее на то, что локальные переменные процедуры сохраняются в промежутках времени между вызовами этой процедуры; `name` — имя процедуры, удовлетворяющее стандартным правилам именованию переменных; `arglist` — список параметров, значения которых передаются в процедуру или возвращаются из процедуры при ее вызове, разделителем в списке параметров является запятая;

statements — любая группа операторов, выполняемых в процедуре; **Exit Sub** — оператор, приводящий к немедленному выходу из процедуры и **End Sub** — оператор, отмечающий конец процедуры.

Процедуру можно вызвать из другой процедуры или функции двумя способами.

□ С помощью указателя:

```
name arglist,
```

где name — имя вызываемой процедуры, а arglist — список аргументов, передаваемых процедуре (он должен соответствовать по количеству и типу списку параметров, заданному в процедуре при ее определении, аргументы в списке разделяются запятыми).

□ С помощью оператора **Call**:

```
Call name( arglist )
```

Обратите внимание, что в этом случае список фактических параметров заключается в скобки. При первом способе скобки не использовались.

Используя именованные параметры, можно вводить аргументы в любом порядке и опускать необязательные. При этом после имени аргумента ставятся двоеточие и знак равенства, после которого помещается значение аргумента. Приводимый далее код показывает основные способы передачи параметров на примере вызова процедуры CircleLength:

```
Private Sub CircleLength( DblR As Double )
    Dim DblPi As Double
    DblPi = Atn( 1# ) * 4#
    MsgBox 2# * DblR * DblPi
End Sub
Private Sub ShowResults( )
    CircleLength DblR:=1# ' Первый способ вызова процедуры
    Dim DblRs As Double
    DblRs = 2#
    CircleLength DblRs      ' Второй способ вызова процедуры
    ' Третий способ вызова процедуры
    Call CircleLength( 3# )
End Sub
```

Кроме процедуры, в языке VBA имеется *функция* **Function**. Синтаксис функции содержит те же элементы, что и процедура. Оператор **Exit Function** приводит к немедленному выходу из функции. Подобно процедуре, функция является самостоятельной процедурой, которая может получать значения аргументов, выполнять последовательность операторов и изменять значения своих аргументов. Однако в отличие от процедуры, когда требуется использовать возвращаемое функцией значение, функция **Function** может применяться в правой части выражения, как и любая другая стандартная функция, например, Cos. Функция **Function** вызывается в выражении по своему имени, за которым следует список параметров, заключенный в скобки. Для возврата значения из функции в ее теле следует присвоить значение имени функции. Любое число таких операторов присваивания может находиться в любом месте функции.

В следующем далее коде приводится пример функции DblSum, которая находит сумму двух своих аргументов:

```

Sub DemoFun( )
    MsgBox DblSum( 1#, 3# )
End Sub
Function DblSum( DblX As Double, DblY As Double ) _
    As Double
    DblSum = DblX + DblY
End Function

```

Передача параметров процедуры или функции по ссылке или значению. При вызове процедуры или функции в нее передаются некоторые аргументы. Если внутри процедуры или функции этим аргументам нужно присвоить какие-то значения, которые нужно сохранять после выхода из процедуры или функции, то в качестве параметров в процедуру или функцию следует передавать физические адреса переменных (использовать передачу параметров по ссылке). Благодаря этому внутри процедуры или функции может быть модифицировано их содержание. Для явного указания передачи параметров в процедуру или функцию по ссылке используется служебное слово **ByRef**. Другим способом передачи параметров в процедуру или функцию является передача их по значению. При этом способе передачи параметра в процедуру попадает не сама переменная, а ее значение (копия аргумента из вызова). Передача параметра по значению задается служебным словом **ByVal** и применяется для передачи исходных данных. В следующем далее примере показано отличие передачи параметра по ссылке от передачи параметра по значению:

```

Sub DemoByValByRef( ByVal IntA As Integer, IntB As _
    Integer, ByRef IntC As Integer )
    ' IntA передается по значению
    ' по умолчанию IntB передается по ссылке
    ' IntC передается по ссылке
    IntA = IntA + 1
    IntB = IntB + IntA
    IntC = IntC + IntA
End Sub
Sub Test( )
    Dim IntA As Integer
    Dim IntB As Integer
    Dim IntC As Integer
    IntA = 1: IntB = 10: IntC = 100
    DemoByValByRef IntA, IntB, IntC
    MsgBox IntA
    ' Отобразится 1
    MsgBox IntB
    ' Отобразится 12
    MsgBox IntC
    ' Отобразится 102
End Sub

```

4.6.1.27. Рекурсивные процедуры и функции

В языке VBA возможно создание рекурсивных процедур и функций, т. е. процедур и функций, вызывающих сами себя. Классическим примером рекурсивной функции является функция, возвращающая очередной член последовательности чисел Фибоначчи. Два первых члена ряда Фибоначчи равны 1, а каждый его последующий член представляет собой сумму двух предыдущих. Таким образом, n -е число Фибоначчи $F_i(n)$ определяется следующим соотношением:

$$F_i(n) = F_i(n-1) + F_i(n-2), \text{ где } F_i(1) = F_i(2) = 1.$$

В следующем далее примере показано использование рекурсивной функции для вычисления очередного числа Фибоначчи:

```
Sub Test( )
    MsgBox LngFi( 4& )
End Sub
Function LngFi( LngN As Long ) As Long
    If LngN=1& Or LngN=2& Then
        LngFi = 1&
    Else
        LngFi = LngFi( LngN-1& ) + LngFi( LngN-2& )
    End If
End Function
```

4.6.1.28. Встроенные (стандартные) диалоговые окна

В проектах на языке VBA часто встречаются две разновидности стандартных окон диалога — окна сообщений и окна ввода. Они встроены в язык и, если их возможностей достаточно, то можно обойтись без проектирования пользовательских окон диалога. Окно сообщений (MsgBox) выводит простейшие сообщения для пользователя, а окно ввода (InputBox) обеспечивает ввод информации.

Функция *InputBox* выводит на экран диалоговое окно, содержащее сообщение, поле ввода и кнопки **ОК** и **Cancel**. Она устанавливает режим ожидания ввода текста пользователем, а затем, при нажатии на кнопку **ОК**, возвращает значение типа String, содержащее текст, введенный в поле ввода. При нажатии кнопки **Cancel** возвращается пустая строка (**Empty**). Функция имеет следующий формат:

```
InputBox( Prompt[, Title][, Default][, Xpos][, Ypos]
    [, Helpfile, Context] )
```

где используются следующие параметры.

- **Prompt** — обязательное строковое выражение, отображаемое как сообщение в диалоговом окне. Строковое выражение Prompt может содержать несколько строк. Для разделения строк допускается использование символа возврата каретки (Chr(13)), символа перевода строки (Chr(10)) или комбинации этих символов (Chr(13)&Chr(10)).
- **Title** — строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот параметр опущен, то в строку заголовка помещается имя приложения.

- ❑ `Default` — строковое выражение, отображаемое в поле ввода как используемое по умолчанию, если пользователь не введет другую строку. Если этот параметр опущен, то поле ввода изображается пустым.
- ❑ `Xpos` — числовое выражение, задающее расстояние в пикселях по горизонтали между левой границей диалогового окна и левым краем экрана. Если этот параметр опущен, то диалоговое окно выравнивается по горизонтали по центру экрана.
- ❑ `Ypos` — числовое выражение, задающее расстояние по вертикали между верхней границей диалогового окна и верхним краем экрана. Если этот параметр опущен, то диалоговое окно помещается по вертикали примерно на одну треть высоты экрана.
- ❑ `Helpfile` — необязательное строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот параметр указан, то необходимо указать также параметр `Context`.
- ❑ `Context` — числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот параметр указан, то необходимо указать также параметр `Helpfile`.

В качестве примера приведем далее код, выполнение которого приведет к появлению на экране диалогового окна ввода, представленного на рис. 4.67:

```

Sub DemoInputBox( )
    Dim StrName As String
    StrName = InputBox( "Введите ваше имя", _
        "Пример окна ввода" )
    MsgBox "Здравствуйте " & StrName & "!"
End Sub

```

При вводе данных при помощи функции `InputBox` разумно в программе предусмотреть как *проверку вводимых данных*, так и сам факт ввода таковых или отказа от ввода путем нажатия кнопки **Cancel**. Следующий далее пример демонстрирует, как это можно сделать при вводе дат. В примере предлагается ввести будущую дату по отношению к текущей. При помощи функции `IsDate` сначала проверяется, является ли введенное данное датой, и если таковым является, оно преобразуется функцией `CDate` к типу **Date** и сравнивается с текущей.

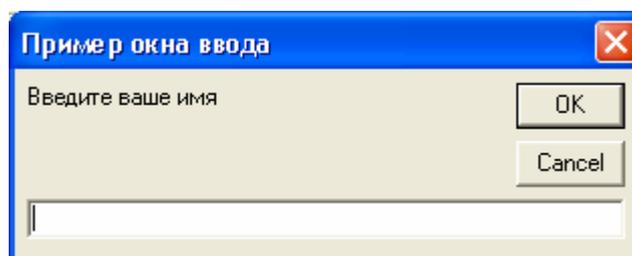


Рис. 4.67. Вид диалогового окна ввода

```

Sub DemoInputWithChecking( )

```

```

Dim VntDate As Variant
VntDate = InputBox( "Введите будущую дату:", _
    "Пример", Date+1 )
' Третий параметр InputBox( ) используется для того,
' чтобы в окне ввода был текст по умолчанию,
' которым в данном случае будет являться следующий
' календарный день
If Not IsDate( VntDate ) Then
    Select Case VntDate
        Case ""
            MsgBox _
                "Нажали кнопку Cancel или ничего не ввели"
        Case Else
            MsgBox "Некорректная дата"
    End Select
Else
    Select Case CDate( VntDate )
        ' Is является значением CDate(VntDate)
        Case Is <= Date
            MsgBox "Должна быть будущая дата"
        Case Is > Date
            MsgBox VntDate & " - введенная дата"
    End Select
End If
End Sub

```

Конечно, если вы уверены в корректности формата вводимых данных, то проверку на тип данных осуществлять не надо, но, тем не менее, позаботиться о проверке того, была ли нажата кнопка **Cancel** или ничего не введено, стоит, например, так, как это делается в следующем далее коде:

```

Sub DemoInputWithChecking1( )
    Dim VntX As Variant
    VntX = InputBox( "Введите значение:", "Пример" )
    If VntX <> "" Then
        MsgBox VntX & " - введенное значение"
    Else
        MsgBox "Нажали Cancel или ничего не ввели"
    End If
End Sub

```

Функция *MsgBox* выводит на экран окно диалога, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа **Integer**, указывающее, какая кнопка была нажата. Функция имеет указанный далее синтаксис

```
MsgBox( Prompt[, Buttons][, Title][, Helpfile, Context] )
```

и содержит следующие параметры.

- Prompt** — обязательное строковое выражение, отображаемое как сообщение в окне диалога.
- Buttons** — числовое выражение, представляющее сумму значений (именованных констант), которые указывают число и тип отображаемых кнопок,

тип используемого значка, основную кнопку и модальность окна сообщения. Значение этого параметра по умолчанию равно 0 (окно диалога содержит только одну кнопку **ОК**). Значения констант, определяющих число, тип кнопок и тип используемого значка приведены в табл. 4.12, 4.13; значения, определяющие основную кнопку окна диалога, приведены в табл. 4.14.

- ❑ `Title` — строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот параметр опущен, то в строку заголовка помещается имя приложения.
- ❑ `Helpfile` — необязательное строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот параметр указан, то необходимо указать также параметр `Context`.
- ❑ `Context` — числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот параметр указан, то необходимо указать также параметр `Helpfile`.

Таблица 4.12. Именованные константы, определяющие кнопки окна диалога (могут комбинироваться в параметре `Buttons`)

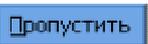
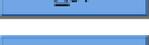
Именованная константа	Значение	Отображаемые кнопки
<code>vbOKOnly, ebOKOnly</code>	0	
<code>vbOKCancel, ebOKCancel</code>	1	 
<code>vbAbortRetryIgnore, ebAbortRetryIgnore</code>	2	  
<code>vbYesNoCancel, ebYesNoCancel</code>	3	  
<code>vbYesNo, ebYesNo</code>	4	 
<code>vbRetryCancel, ebRetryCancel</code>	5	 

Таблица 4.13. Именованные константы, определяющие информационные значки окна диалога (могут комбинироваться в параметре `Buttons`)

Именованная константа	Значение	Отображаемые значки
<code>vbCritical, ebCritical</code>	16	
<code>vbQuestion, ebQuestion</code>	32	
<code>vbExclamation, ebExclamation</code>	48	



Таблица 4.14. Именованные константы, определяющие основную кнопку окна диалога (могут комбинироваться в параметре *Buttons*)

Именованная константа	Значение	Номер основной кнопки
vbDefaultButton1, ebDefaultButton1	0	1
vbDefaultButton2, ebDefaultButton2	256	2
vbDefaultButton3, ebDefaultButton3	512	3
vbDefaultButton4, ebDefaultButton4	768	4

При написании программ с откликом в зависимости от того, какая кнопка окна диалога нажата, вместо возвращаемых значений удобнее использовать именованные константы, перечисленные в табл. 4.15. Их использование делает код программы более прозрачным для чтения и, к тому же, именованные константы легче запомнить.

Таблица 4.15. Именованные константы, идентифицирующие нажатую кнопку

Именованная константа	Значение	Нажатая кнопка
vbOK, ebOK	1	ОК
vbCancel, ebCancel	2	Отмена (Cancel)
vbAbort, ebAbort	3	Прервать (Abort)
vbRetry, ebRetry	4	Повторить (Retry)
vbIgnore, ebIgnore	5	Пропустить (Ignore)
vbYes, ebYes	6	Да (Yes)
vbNo, ebNo	7	Нет (No)

Проиллюстрируем использование окон сообщений. В приводимом далее примере вычисляется квадрат введенного значения 2, и найденное значение затем последовательно отображается в пяти окнах сообщения. Первое окно является простым окном с сообщением (рис. 4.68), второе — с сообщением, выводимым в две строки (рис. 4.69), третье — с сообщением и информационным значком (рис. 4.70), четвертое — с сообщением, информационным значком и двумя кнопками **Да** и **Нет**, причем кнопка **Да** установлена как кнопка, выполняемая по умолчанию (рис. 4.71), а пятое — с сообщением, информационным значком и пользовательским заголовком (рис. 4.72):

```
Sub DemoMsgBox( )
    Dim VntX As Variant
    Dim DblY As String
    VntX = InputBox( "Введите x" )
    DblY = VntX^2
    MsgBox "VntX="+VntX
    MsgBox "VntX="+VntX+Chr(13)+Chr(10)+"DblY=VntX^2="+DblY
    MsgBox "DblY="&DblY, ebInformation
```

```

MsgBox "DbY=" & DbY, _
      ebInformation+ebDefaultButton1+ebYesNo
MsgBox "DbY=" & DbY, ebInformation, _
      "Нахождение квадрата"
End Sub

```



Рис. 4.68. Простое окно сообщения

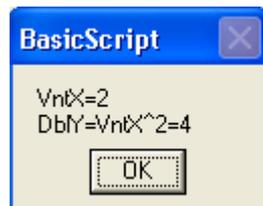


Рис. 4.69. Окно с сообщением, выводимым в две строки

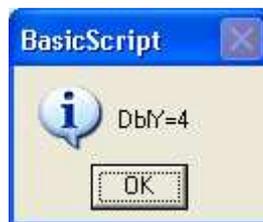


Рис. 4.70. Окно с сообщением и информационным значком



Рис. 4.71. Окно с сообщением, информационным значком и кнопками **Да** (по умолчанию) и **Нет**

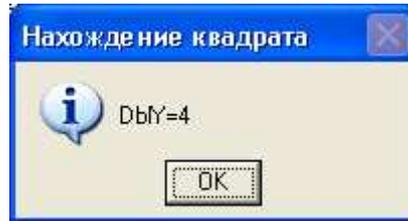


Рис. 4.72. Окно с сообщением, информационным значком и пользовательским заголовком

Определение кнопки, выбранной в окне сообщения. Функция MsgBox удобна для вывода той или иной информации. Однако если надо узнать, какой выбор сделал пользователь при нажатии отображаемых в диалоговом окне кнопок, то процедуру MsgBox надо использовать не как процедуру, а как функцию. В этом случае значение, возвращаемое функцией MsgBox, надо присваивать какой-то переменной, а ее параметры заключать в скобки.

Рассмотрим далее пример использования функции MsgBox с анализом нажатой кнопки. В приводимом примере на экране отображается диалоговое окно с двумя кнопками **Да** (выбрана по умолчанию), **Нет** и вопросительным знаком. При нажатии кнопки **Да** (нажатии клавиши **Enter**) выдается сообщение с текущей датой, а при нажатии кнопки **Нет** — пустое окно:

```
Sub DemoYesNo( )
    Dim IntAnswer As Integer
    IntAnswer = MsgBox( "Показать текущую дату?", _
        vbYesNo + vbQuestion + vbDefaultButton1 )
    Select Case IntAnswer
        Case vbYes
            MsgBox Now          ' Текущая дата и время
        Case vbNo
            MsgBox "", vbExclamation
    End Select
End Sub
```

4.6.1.29. Стандартные функции и операторы для работы с файлами

В языке VBA для работы с файлами можно использовать ряд стандартных функций и операторов, которые можно разделить на *следующие группы*, выполняющие:

- получение информации о файле EOF, FileDateTime, FileLen, FreeFile, Loc, LOF;
- управление файлами Dir, Kill, Name, FileCopy;
- задание или получение атрибутов файла FileAttr, GetAttr, SetAttr;
- открытие файла Open;
- получение или установку в файле позиции чтения-записи Seek;
- чтение данных из текстового файла Input, Input #, Line Input #;
- запись данных в текстовый файл Print #, Write #;

□ закрытие файла Close.

4.6.1.29.1. Получение информации о файле

Функция EOF возвращает значение **True** при достижении конца открытого файла, **False** — в противном случае и имеет следующий синтаксис:

```
EOF( filenumber )
```

Здесь *filenumber* — номер файла, который был использован в операторе Open при открытии файла.

Функция FileDateTime возвращает значение **Variant(Date)**, представляющее дату и время создания или последней модификации файла, и имеет следующий синтаксис:

```
FileDateTime( pathname )
```

Здесь *pathname* — строка, специфицирующая имя файла (может содержать полный путь, включая диск и папки). Приведем далее пример использования этой функции:

```
Sub DemoFileDateTime( )
    Dim VntMyStamp
    ' Файл t.txt модифицирован 7 января 2007 в 14:13:31
    VntMyStamp = FileDateTime( "t.txt " )
    ' Возвращает 7.01.2007 14:13:31
End Sub
```

Функция FileLen возвращает размер с типом **Long** указанного файла в байтах (до открытия файла) и имеет следующий синтаксис:

```
FileLen( pathname )
```

Здесь *pathname* — строка, специфицирующая имя файла (может содержать полный путь, включая диск и папки).

Функция LOF возвращает размер с типом **Long** открытого файла в байтах и имеет следующий синтаксис:

```
LOF( filenumber )
```

Здесь *filenumber* — номер файла, который был использован в операторе Open при открытии файла.

Функция FreeFile возвращает значение **Integer**, представляющее свободный последний номер открытого файла, и имеет следующий синтаксис:

```
FreeFile[( Rangenumber )]
```

Здесь *Rangenumber* — необязательный аргумент, определяющий диапазон для возвращаемого номера файла (при аргументе 0 диапазон от 0 до 255, соответствует умолчанию; при аргументе 1 диапазон от 256 до 511).

Функция Loc возвращает значение **Long**, представляющее текущую позицию чтения-записи в открытом файле, и имеет следующий синтаксис:

```
Loc( filenumber )
```

Здесь *filenumber* — номер файла, который был использован в операторе Open при открытии файла.

4.6.1.29.2. Управление файлами

Функция *Dir* возвращает значение **String**, представляющее имя файла, каталога или диска, соответствующее указанной маске или атрибуту файла, и имеет следующий синтаксис:

```
Dir([pathname[, attributes]])
```

Здесь *pathname* — последовательность символов, обозначающая путь или имя файла (она может содержать маски и обозначения дисков); *attributes* — задает атрибуты искомого файлов (скрытый, системный или метка тома). Если последний аргумент не указан, то функция возвращает файлы без атрибутов, соответствующие маске *pathname*. Если опущены все аргументы, то функция возвращает очередное для текущего каталога имя подкаталога, служебное имя "." или ".." или имя файла. В качестве аргумента *attributes* могут использоваться комбинации атрибутов, приведенных в табл. 4.16.

Таблица 4.16. Атрибуты файлов

Именованная константа	Значение	Описание
vbNormal	0	Обычный
vbHidden	2	Скрытый
vbSystem	4	Системный
vbVolume	8	Метка тома
vbDirectory	16	Каталог

Приведем далее пример использования этой функции:

```
sub DemoDir( )
    Dim MyFile, MyPath, MyName
    ' При наличии возвращает "WIN.INI"
    MyFile = Dir( "C:\WINDOWS\WIN.INI" )
    ' Возвращает первый из файлов с расширением .ini из
    ' указанного каталога
    MyFile = Dir( "C:\WINDOWS\*.INI" )
    ' Возвращает первый *.TXT файл из текущего каталога
    MyFile = Dir( "*.TXT" )
    ' Отображает каталоги диска C:\
    MyPath = "c:\"
    MyName = Dir( MyPath, vbDirectory ) ' Первый каталог
    Do While MyName <> "" ' Найдено непустое имя
        If MyName <> "." And MyName <> ".." Then
            ' Найденное имя не является служебными именем
            ' "." или "..". О GetAttr( ) см. далее в
            ' подразд. 1.29.3
            If ( GetAttr( MyPath & MyName ) And _
                vbDirectory ) = vbDirectory Then
                Debug.Print MyName
            End If
        End If
    End While
    MyName = Dir
```

Loop

End Sub

Оператор Kill удаляет указанный аргументом файл и имеет следующий синтаксис:

```
Kill pathname
```

Здесь *pathname* — строка, специфицирующая имя удаляемого файла (может содержать полный путь, включая диск и папки, и содержать маски * и ?). Приведем далее пример использования этого оператора:

```
Sub DemoKill( )
  Kill "TestFile" ' Удаление файла в текущем каталоге
  ' Удаление всех *.TXT файлов в текущем каталоге
  Kill "*.TXT"
End Sub
```

End Sub

Оператор Name переименовывает файл или каталог и имеет следующий синтаксис:

```
Name oldpathname As newpathname
```

Здесь *pathname* — строка, специфицирующая имя переименоваемого файла (может содержать полный путь, включая диск и папки); *newpathname* — строка, специфицирующая имя нового файла (может содержать полный путь, включая диск и папки). Приведем далее пример использования этого оператора:

```
Sub DemoName( )
  Dim OldName, NewName
  OldName = "OLDFILE": NewName = "NEWFILE"
  Name OldName As NewName ' Переименование файла
  OldName = "C:\TMP\OLDFILE"
  NewName = "C:\TMP1\NEWFILE"
  ' Перемещение и переименование файла
  Name OldName As NewName
End Sub
```

End Sub

Оператор FileCopy копирует файл и имеет следующий синтаксис:

```
FileCopy source, destination
```

Здесь *source* — строка, специфицирующая имя копируемого файла (может содержать полный путь, включая диск и папки); *destination* — строка, специфицирующая новое имя файла (может содержать полный путь, включая диск и папки). Приведем далее пример использования этого оператора:

```
Sub DemoFileCopy( )
  Dim SourceFile, DestinationFile
  SourceFile = "NEWFILE"
  DestinationFile = "COPY"
  FileCopy SourceFile, DestinationFile
End Sub
```

End Sub

4.6.1.29.3. Задание или получение атрибутов файла

Функция FileAttr возвращает значение **Long**, представляющее режим, в котором находится открытый файл (табл. 4.17), и имеет следующий синтаксис:

```
FileAttr( filenumber, returntype )
```

Здесь *filenumber* — номер файла, который был использован при открытии файла оператором *Open*; *returntype* — имеет значение 2 для 32-разрядных систем.

Таблица 4.17. Значения, возвращаемые функцией FileAttr

Возвращаемое значение	Режим файла
1	Input
2	Output
4	Random
8	Append
32	Binary

Приведем далее пример использования этой функции:

```
Sub DemoFileAttr( )
  Dim FileNum, Mode
  FileNum = 1
  Open "t.txt" For Append As FileNum
  Mode = FileAttr( FileNum, 1 )      ' Возвращает 8
  Close FileNum
End Sub
```

Функция *GetAttr* возвращает значение **Integer**, представляющее атрибуты файла, каталога или диска (табл. 4.18). Если заданный файл имеет несколько атрибутов, то возвращается сумма соответствующих отдельным атрибутам значений. Функция имеет следующий синтаксис:

```
GetAttr( pathname )
```

Здесь *pathname* — специфицирует файл, каталог или диск.

Таблица 4.18. Атрибуты файлов

Именованная константа	Значение	Описание
vbNormal	0	Обычный
vbReadOnly	1	Только для чтения
vbHidden	2	Скрытый
vbSystem	4	Системный
vbVolume	8	Метка тома
vbDirectory	16	Каталог
vbArchive	32	Архивный

Оператор *SetAttr* устанавливает атрибуты файла и имеет следующий синтаксис:

```
SetAttr pathname, attributes
```

Здесь *pathname* — строка, специфицирующая имя файла (может содержать полный путь, включая диск и папки); *attributes* — специфицирует атрибуты файла (табл. 4.18, если заданный файл имеет несколько атрибутов, то задается сумма соответствующих отдельным атрибутам значений).

4.6.1.29.4. Открытие файла

Оператор *Open* выполняет открытие файла и имеет следующий синтаксис:

```
Open pathname For mode [Access access] [lock] As
```

```
[#]filename [Len=reclength]
```

где:

- `pathname` — путь к файлу и его полное имя.
- `mode` — режим доступа к файлу (Append: запись в конец; Output: запись в начало; Binary: двоичный доступ; Input: чтение; Random: чтение-запись, по умолчанию).
- `access` — тип доступа к файлу, необязательный параметр (Read: чтение, Write: запись, Read Write: чтение-запись).
- `lock` — необязательный параметр, определяющий совместный доступ к файлу (Shared: совместный доступ, Lock Read: блокировка чтения, Lock Write: блокировка записи, Lock Read Write: блокировка чтения и записи).
- `filename` — номер файла от 1 до 511.
- `reclength` — необязательный параметр, определяет размер буфера (не более 32767).

Пример использования оператора **Open** приведен ранее в описании функции FileAttr.

4.6.1.29.5. Получение или установка в файле позиции чтения-записи

Функция *Seek* позволяет получить или установить в файле позицию чтения-записи и может использоваться в двух синтаксических формах.

- Получение из файла позиции чтения-записи:

```
Seek( filename )
```

Здесь `filename` — номер файла, открытого оператором **Open**. Функция возвращает значение **Long** в диапазоне от 1 до $2^{31}-1$, специфицирующее текущую позицию чтения-записи.

- Установка в файле позиции чтения-записи:

```
Seek [#]filename, position
```

Здесь `filename` — номер файла, открытого оператором **Open**; `position` — значение **Long** в диапазоне от 1 до $2^{31}-1$, специфицирующее следующую текущую позицию чтения-записи.

4.6.1.29.6. Чтение данных из текстового файла

Особенность *текстовых файлов* заключается в способе доступа к данным, которые в них содержатся. Текстовый файл является файлом последовательного доступа, в котором содержится последовательность ASCII-символов. Текстовый файл может быть форматированным, то есть содержать пробелы, табуляторы, символы перехода на новую строку и другие специальные знаки. Чтение данных из текстового файла осуществляется с помощью функции *Input*, а также операторов **Input #** и **Line Input #**.

Функция *Input* является низкоуровневой и позволяет последовательно прочитать заданное в ней количество символов. Функция имеет следующий синтаксис:

```
Input( number, [#]filename )
```

Здесь `number` — целочисленный аргумент, определяющий число читаемых символов, начиная с текущего (вначале текущим является первый символ файла); `filenumber` — номер файла, открытого с помощью оператора **Open**.

Оператор Input # позволяет читать форматированные текстовые файлы, где в качестве разделителя используются запятые. Оператор имеет следующий синтаксис:

```
Input #filenumber, varlist
```

Здесь `filenumber` — номер файла, открытого с помощью оператора **Open**; `varlist` — список переменных, разделенных запятыми, которым последовательно присваиваются читаемые данные.

Оператор Line Input # выполняет посимвольное чтение очередной строки текстового файла. Символы окончания строки (CHR(13) или CHR(13)+ CHR(10)) в прочитанную строку не включаются. Оператор имеет следующий синтаксис:

```
Line Input #filenumber, varname
```

Здесь `filenumber` — номер файла, открытого с помощью оператора **Open**; `varname` — имя переменной с типом **Variant** или **String**, в которую помещается прочитанная строка. При выполнении этого оператора проверка достижения конца файла не производится. Поэтому для проверки достижения конца файла следует использовать функцию EOF.

В приведенном далее иллюстрирующем примере выполняется построчное считывание данных текстового файла с их выводом в окно **Immediate**, а затем читаются и выводятся в это же окно еще пять символов этого же файла, начиная с третьего от начала:

```
Sub Demo( )
  Dim StrLine
  Open "text.txt" For Input As #1
  On Error GoTo ErLabel:      ' Обработка ошибки
  While EOF( 1 ) = False
    Line Input #1, StrLine
    Debug.Print StrLine
  Wend
  Seek #1, 3
  Debug.Print Input( 5, #1 )
  Close #1
  Exit Sub
ErLabel:
  MsgBox "Ошибка!"
End Sub
```

4.6.1.29.7. Запись данных в текстовый файл

Запись в текстовый файл выполняется с помощью операторов **Print #** и **Write #**. Запись всегда производится в файл, открытый для записи (режим `Output`) или для добавления (режим `Append`). В первом случае данные, содержащиеся в файле, уничтожаются, а во втором — запись происходит в конец файла.

При использовании *оператора Write #* вывод происходит в одну строку, выводимые данные разделяются запятыми, а в конце строки вставляются символы перевода каретки и новой строки. Оператор имеет следующий синтаксис:

```
Write #filename, [outputlist]
```

Здесь *filename* — номер файла, открытого с помощью оператора **Open**; *outputlist* — имя переменной или список переменных, разделенных запятыми (разделителями могут быть также пробелы и точки с запятой). Приведем далее пример использования этого оператора:

```
Sub Demo( )
  ' Открытие файла для записи
  Open "TESTFILE" For Output As #1
  Write #1, "Hello World", 234 ' Разделитель - запятая
  Write #1, ' Запись пустой строки
  Dim MyBool, MyDate, MyNull, MyError
  MyBool = False: MyDate = #2/12/1969#: MyNull = Null
  MyError = CVer( 32767 )
  Write #1, MyBool; " is a Boolean value"
  Write #1, MyDate; " is a date"
  Write #1, MyNull; " is a null value"
  Write #1, MyError; " is an error value"
  Close #1 ' Закрытие Файла
End Sub
```

В результате содержимое файла TESTFILE получает следующий вид:

```
"Hello World",234
```

```
#FALSE#," is a Boolean value"
```

```
#1969-02-12#," is a date"
```

```
#NULL#," is a null value"
```

```
#ERROR 32767#," is an error value"
```

Оператор Print # является более гибким оператором записи в файл и имеет следующий синтаксис:

```
Print #filename, [{Spc(n)|Tab[(n)]}] [expression]
[charpos]
```

Здесь *filename* — номер файла, открытого с помощью оператора **Open**; *Spc(n)|Tab[(n)]* — опция, позволяющая указать тип (пробел или табулятор) и число (n) разделителей между значениями переменных; *expression* — список выражений, разделяемых запятыми, значения которых выводятся в файл; *charpos* — тип разделителя между выводимыми значениями (по умолчанию — точка с запятой). Если аргумент *charpos* опущен, то происходит переход к следующей строке. Приведем далее пример использования этого оператора:

```
Sub Demo( )
  Open "TESTFILE" For Output As #1
  Print #1, "This is a test"
  Print #1,
  Print #1, "Zone 1"; Tab; "Zone 2"
  Print #1, "Hello"; " "; "World"
```

```

Print #1, Spc( 5 ); "5 leading spaces "
Print #1, Tab( 10 ); "Hello"
Dim MyBool, MyDate, MyNull, MyError
MyBool = False: MyDate = #2/12/1969#: MyNull = Null
MyError = CVErr( 32767 )
Print #1, MyBool; " is a Boolean value"
Print #1, MyDate; " is a date"
Print #1, MyNull; " is a null value"
Print #1, MyError; " is an error value"
Close #1

```

End Sub

В результате содержимое файла TESTFILE получает следующий вид:

This is a test

Zone 1 Zone 2

Hello World

 5 leading spaces

 Hello

False is a Boolean value

12.02.1969 is a date

Null is a null value

Error 32767 is an error value

4.6.1.29.8. Закрытие файла

После окончания работы с открытым файлом его необходимо корректно закрыть. Автоматическое закрытие всех открытых файлов происходит при завершении работы приложения. Однако, при этом возможно возникновение различных неконтролируемых сбоев, что может привести к искажению файлов. Кроме того, незакрытый файл увеличивает требования к ресурсам.

Оператор Close предназначен для закрытия ранее открытого файла и имеет следующий синтаксис:

Close [filenumberlist]

Здесь *filenumberlist* — номер или список разделенных запятыми номеров файлов, которые надо закрыть (каждому номеру необязательно может предшествовать символ #). Если этот аргумент опущен, то будут закрыты все открытые ранее файлы.

Примечание

Наряду с перечисленными ранее стандартными функциями и операторами для работы с файлами можно использовать объекты и коллекции объектов языка VBA, такие, как File Object, Files Collection, FileSystemObject Object, Folder Object, Folders Collection и TextStreamObject. Для получения справочной информации о них достаточно в интегрированной среде разработки (ИСП) VBA-программ в окне редактирования кода поместить курсор на любое служебное слово и нажать клавишу **F1**. В появившемся окне **Visual Basic Reference** следует выбрать вкладку **Содержание**, открыть раздел **Objects** и посмотреть темы File Object, Files Collection, FileSystemObject Object, Folder Object, Folders Collection и TextStreamObject. Дополнительную информацию о перечисленных ранее объектах и их коллекциях можно найти в [15, глава 6] и [14].

4.6.2. Использование языка VBA в SCADA-системе GeniDAQ

VBA-подобный язык, используемый в SCADA-системе GeniDAQ, представляет собой мощное средство программирования, совместимое с языком Microsoft Visual Basic для приложений и лицензированное у компании Summit Software Inc. Он является важным компонентом SCADA-системы GeniDAQ, обеспечивающим возможность разработки специализированных фрагментов приложения на языке программирования высокого уровня. Поскольку языки Visual Basic и их диалекты являются одними из самых популярных, простых для изучения и использования языков программирования, то наличие диалекта языка VBA в SCADA-системе GeniDAQ позволяет значительно упростить и повысить эффективность программирования.

Синтаксис используемой в SCADA-системе GeniDAQ версии языка VBA практически совместим с синтаксисом языков Microsoft VBA (используется, например, в приложениях Excel, Word, Access и т. п.) и Microsoft Visual Basic. Хотя версия языка VBA SCADA-системы GeniDAQ и имеет ряд несовместимых функций, но более 95% функций и процедур абсолютно идентичны. Поэтому в SCADA-системе GeniDAQ можно использовать программы, написанные на языке Visual Basic практически без изменений. Номера ошибок и сообщения об ошибках в VBA-программах SCADA-системы GeniDAQ также совместимы с имеющимися в языке Visual Basic. Специализированные, несовместимые функции и процедуры, введенные в язык VBA для привязки к SCADA-системе, обеспечивает не только привязку, но и возможность манипуляции задачами, входящими в приложение, взаимодействия с операционными системами Windows и другими приложениями посредством технологий DDE, OLE, ODBC (SQL) и COM-DCOM (OPC-серверы). Сказанное позволяет заключить, что при разработке VBA-программ для SCADA-системы GeniDAQ можно воспользоваться многочисленной литературой по языку VBA и приведенным ранее описанием этого языка.

Замечание

В SCADA-системе GeniDAQ используется более ранняя версия языка VBA, чем описанная ранее и в современной литературе. Поэтому для уточнения имеющихся немногочисленных отличий целесообразно использовать встроенную в ИСП VBA-программ справочную систему, которую можно активизировать командой **Help | Help Topics** или путем нажатия клавиши **F1**.

Далее применительно к SCADA-системе GeniDAQ последовательно рассматриваются ИСП VBA-программы, встроенная в ИСП справочная система языка VBA, структура программного проекта приложения и средства привязки VBA-программы к использующему ее SCADA-приложению.

4.6.2.1. Интегрированная среда разработки VBA-программы

Вид ИСП VBA-программы показан на рис. 4.73. *Интегрированная среда разработки VBA-программы* представляет собой текстовый редактор с рядом удобных возможностей для редактирования и отладки кода VBA-программы. Исходный текст

будет компилироваться в псевдокод сразу после редактирования, так что не будет необходимости компилировать его вновь при запуске программы. Таким образом, ИСР представляет собой мощное средство разработки VBA-процедур обработки данных, специфичных для задачи приложения. Окно ИСР является модальным, что требует его закрытия перед продолжением работы.

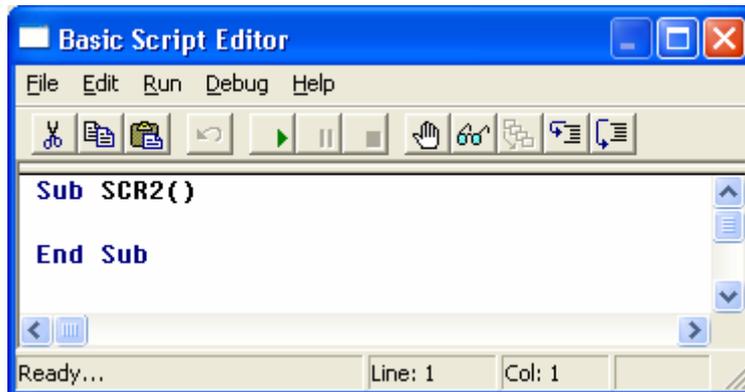


Рис. 4.73. Вид ИСР VBA-программ

В окне ИСР имеется *панель инструментов*, содержащая следующие кнопки:

 (вырезать, cut) — удалить выделенный фрагмент текста и поместить его в универсальный буфер обмена Clipboard, действие этой кнопки эквивалентно выполнению команды **Edit | Cut** или активизации акселератора **Ctrl+X**;

 (скопировать, copy) — поместить копию выделенного фрагмента текста в универсальный буфер обмена Clipboard, действие этой кнопки эквивалентно выполнению команды **Edit | Copy** или активизации акселератора **Ctrl+C**;

 (вставить, paste) — поместить содержимое универсального буфера обмена Clipboard в текст, начиная от текущей позиции курсора, действие этой кнопки эквивалентно выполнению команды **Edit | Paste** или активизации акселератора **Ctrl+V**;

 (отменить, undo) — отменить результат последней выполненной операции редактирования, действие этой кнопки эквивалентно выполнению команды **Edit | Undo** или активизации акселератора **Ctrl+Z**;

 (исполнить, start) — начать исполнение программы, действие этой кнопки эквивалентно выполнению команды **Run | Start** или нажатию клавиши **F5**;

 (прервать, break) — прервать исполнение программы и поместить указатель команд на следующую строку, подлежащую исполнению;

 (остановить, end) — остановить исполнение программы, действие этой кнопки эквивалентно выполнению команды **Run | End**;

 (точка останова, toggle breakpoint) — добавить или удалить точку останова в текст программы, действие этой кнопки эквивалентно выполнению команды **Debug | Toggle Breakpoint** или нажатию клавиши **F9**;

 (контрольное значение, add watch) — вывести диалоговую панель **Add Watch** для просмотра значений переменных, действие этой кнопки эквивалентно выполнению команды **Debug | Add Watch...** или активизации акселератора **Shift+F9**;

 (вызовы, calls) — вывести на экран список процедур, вызванных текущей исполняющейся процедурой (кнопка доступна только после прерывания исполнения программы с помощью команды **прервать**);

 (следующая инструкция, single step) — выполнить инструкцию в следующей строке процедуры и прервать исполнение (если в строке имеется вызов другой процедуры, то исполнение будет продолжено со следующей инструкции вызванной процедурой), действие этой кнопки эквивалентно выполнению команды **Debug | Single Step** или нажатию клавиши **F8**;

 (следующая процедура, procedure step) — выполнить инструкцию в следующей строке процедуры и прервать исполнение (если в строке имеется вызов другой процедуры, то данная процедура будет выполнена целиком), действие этой кнопки эквивалентно выполнению команды **Debug | Procedure Step** или активизации акселератора **Shift+F8**.

При помещении курсора в область отображения кнопки панели инструментов и удержании его в неподвижном состоянии в течение около 1 секунды в область ниже курсора будет выведено краткое описание данной кнопки (всплывающая подсказка). В нижней строке окна — строке статуса — при этом будет выдана более подробная справка о действии кнопки.

Для поиска синтаксических ошибок в тексте процедуры можно использовать команду **Run | Syntax Check**. На экран монитора будет выведено сообщение, указывающее на наличие или отсутствие ошибок в тексте процедуры. Сообщение об ошибке относится к первой строке, в которой была обнаружена ошибка, и содержит краткое пояснение причины, по которой синтаксис в указанной строке признан неверным. При нажатии на кнопку **ОК** в диалоговом окне сообщения об ошибке или нажатии клавиши **Enter** строка процедуры, в которой обнаружена синтаксическая ошибка, будет выделена в окне редактирования. При попытке запуска на исполнение или отладки процедуры, для которой не была проведена проверка синтаксиса, указанная проверка будет выполнена автоматически перед началом исполнения.

При завершении работы ИСП (команда **File | Close**, акселератор **Ctrl+W**) результат выполнения операции зависит от того, были ли внесены какие-либо изменения в исходный текст процедуры, а также от наличия синтаксических ошибок в исходном тексте. Если в процессе работы в исходный текст процедуры были внесены какие-либо изменения, на экран монитора будет выведена диалоговое окно с запросом сохранения выполненных изменений. При этом, если исходный текст сценария содержит синтаксические ошибки и выбрана (нажата) кнопка **Да** диалогового окна, на экран будет выведено окно, содержащее сообщение о наличии ошибок и запрос на подтверждение намерения пользователя продолжить сохранение программы.

Нажатие кнопки **Да** данного диалогового окна приведет к сохранению исходного текста сценария вместе со всеми имеющимися ошибками. Если в исходный текст сценария не было внесено никаких изменений, произойдет мгновенное завершение работы ИСР независимо от наличия ошибок в исходном тексте процедуры, допущенных во время предыдущего сеанса работы.

В окне ИСР с помощью команды **Help | Help Topics** можно получить справочную информацию о языке, используемом для создания VBA-программы.

4.6.2.2. Структура VBA-программы (программный проект приложения)

В общем случае, VBA-программа (программный проект приложения) содержит следующие компоненты.

- *Главная процедура* (главный сценарий, Main Script) программного проекта приложения, будучи включенной в программный проект, осуществляет управление исполнением всего приложения один раз при его запуске. Вид ИСР главной процедуры показан на рис. 4.74. Для включения в программный проект приложения главной процедуры достаточно выполнить в главном окне SCADA-системы команду **File | Add/Delete | Add Main Script** (рис. 4.75). Исключение из программного проекта приложения главной процедуры выполняется командой **File | Add/Delete | Delete Main Script** (см. рис. 4.75). Главная процедура может быть применена для выполнения таких операций, как запуск, останов задач приложения и т.п. Главная процедура является необязательным элементом программного проекта приложения.

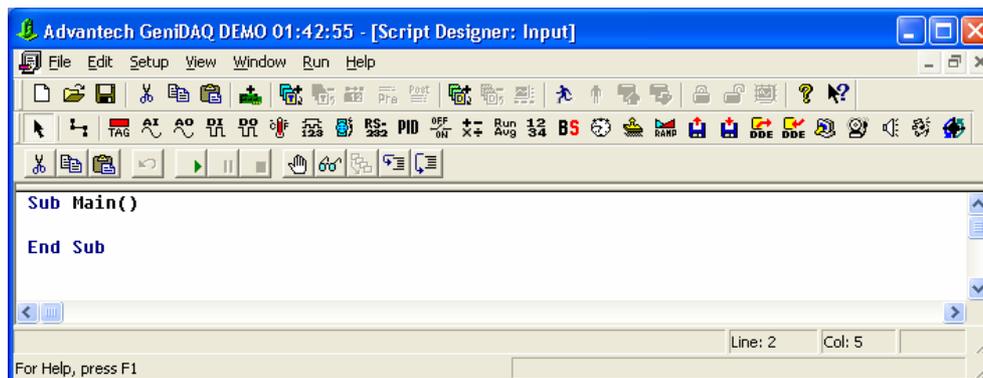


Рис. 4.74. Вид ИСР главной процедуры

- *Предварительная процедура (Pre-Task Script) и пост-процедура (Post-Task Script) задачи.* Каждая сканируемая задача приложения имеет предварительную процедуру и пост-процедуру. Предварительная процедура выполняется один раз перед первым запуском задачи, а пост-процедура — один раз после последнего завершения задачи. Для их редактирования достаточно выполнить команду **Setup | Pre task Script** или команду **Setup | Post task Script** (рис. 4.76 и 4.77). По

умолчанию эти процедуры не выполняют никаких действий (см. рис. 4.76 и 4.77). Указанные два типа процедур при определенных условиях используются для инициализации или сброса значений, связанных с объектами, которые входят в приложения.

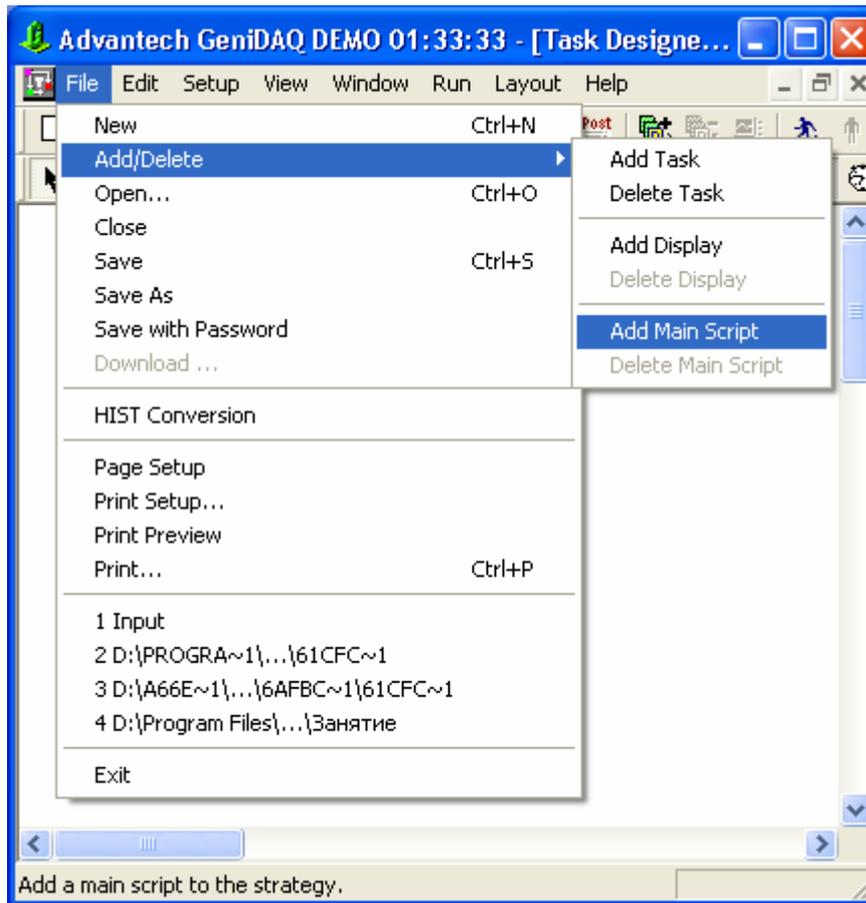


Рис. 4.75. Включение главной процедуры в программный проект приложения

- *Бейсик-процедура* (Бейсик-процедура, Basic Script) автоматически включается в программный проект при включении в задачу приложения функционального блока **BASIC Script**. Таким образом, число Бейсик-процедур программного проекта совпадает с общим числом функциональных блоков **BASIC Script** в приложении. Использование Бейсик-процедур обеспечивает максимальную гибкость при программировании специализированных вычислительных процедур и процедур сбора данных и управления. Вид ИСП Бейсик-процедуры показан ранее на рис. 4.73.

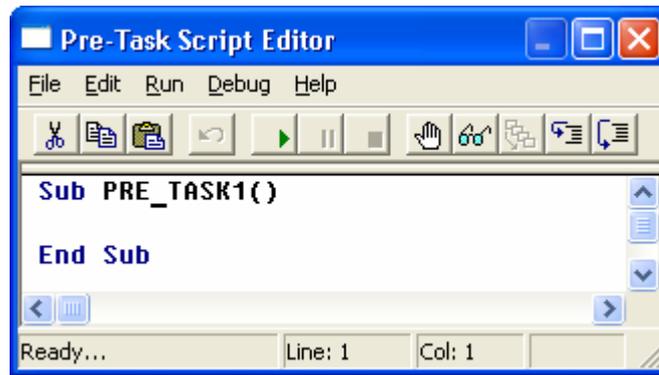


Рис. 4.76. ИСП предварительной процедуры задачи программного проекта приложения

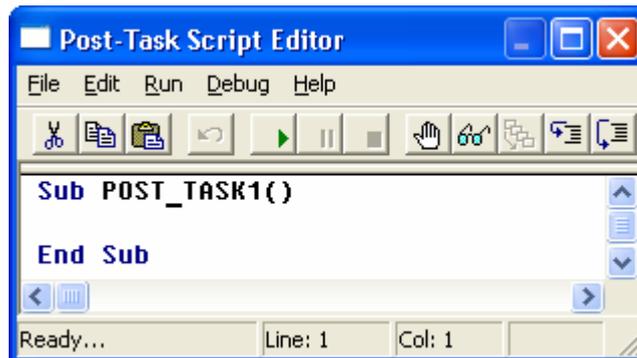


Рис. 4.77. ИСП пост-процедуры задачи программного проекта приложения

4.6.2.3. Справочная система ИСП VBA-программ

Как было указано ранее, *справочную информацию* о языке, используемом для создания VBA-программы, можно получить в ИСП с помощью команды **Help | Help Topics** (рис. 4.78).

При активизации гиперссылки **Command Reference** получаем алфавитно-упорядоченный список гиперссылок на справки об элементах языка BasicScript Language (рис. 4.79).

Аналогичным образом, при активизации гиперссылок **Directives ... Objects** получаем соответствующие алфавитно-упорядоченные списки гиперссылок на справки об элементах языка BasicScript Language по функциональным группам. И, наконец, при активизации гиперссылки **BasicScript Help Contents** получаем окно справки, представленное на рис. 4.80. Если в этом окне активизировать гиперссылку **Dialog Editor**, можно получить алфавитно-упорядоченный список гиперссылок на справки о редакторе диалога (рис. 4.81).

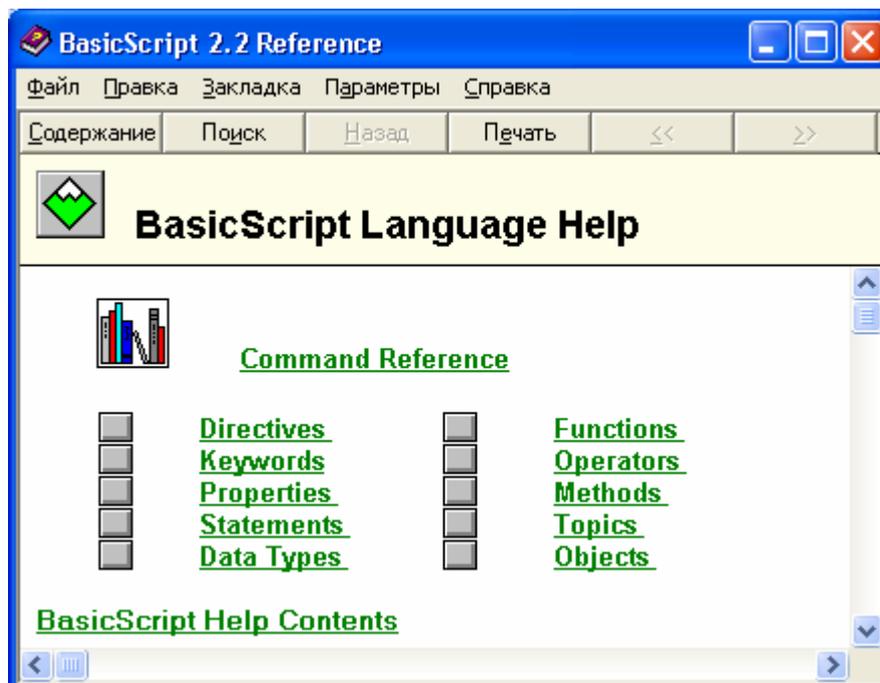


Рис. 4.78. Окно справки ИСП VBA-программ

4.6.2.4. Привязка VBA-программы к использующему ее приложению

Для реализации управления задачами и данными в процессе работы приложения в язык VBA включен ряд специализированных средств. К их числу относятся средства управления задачами, работы с тегами, управления окнами экранных форм и работы с Бейсик-процедурами.

4.6.2.4.1. Управление задачами

Средства управления задачами предназначены для управления сканируемыми задачами, входящими в приложение. Указанные средства могут применяться в главной процедуре, а также в предварительной процедуре и пост-процедуре задачи (задач). К средствам управления задачами относятся следующие.

Тип ScanTask — тип объекта *сканируемая задача*. Этот тип используется для определения переменной, которая позволяет получить доступ к объекту, возвращаемому функцией `GetScanTask()`:

```
' Пример 4.1
' В основном сценарии стратегии выполняется запуск задачи
' TASK1
Sub Main( )
    Dim MyTask As ScanTask
```

```

Set MyTask = GetScanTask( "TASK1" )
MyTask.Start
End Sub

```

См. также GetScanTask (функция), Start (метод).

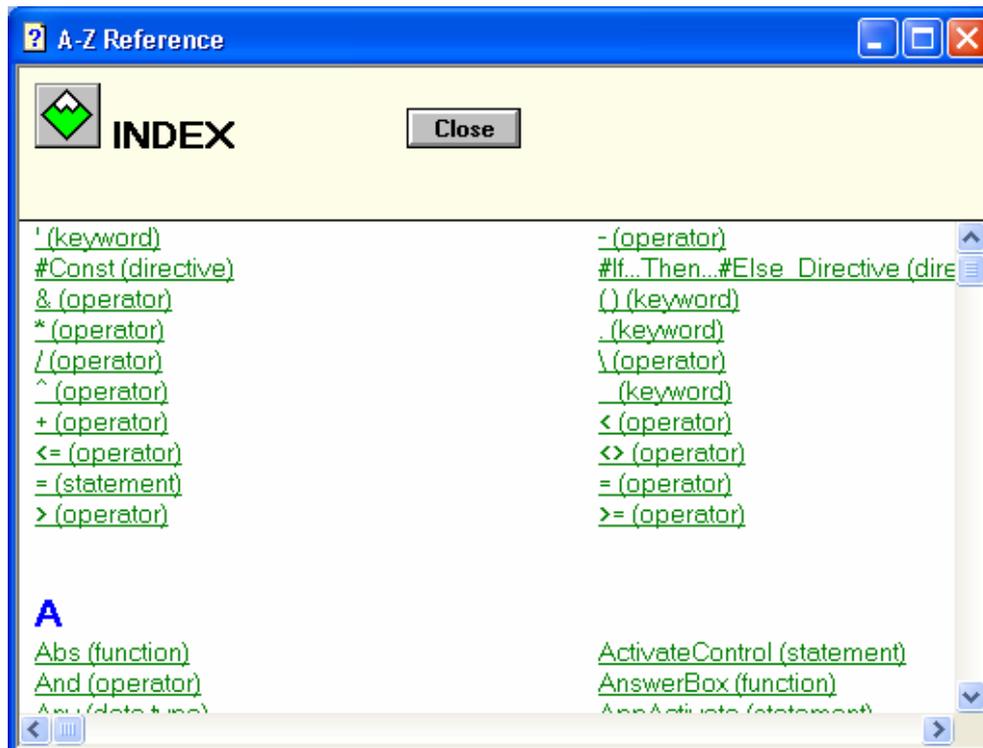


Рис. 4.79. Алфавитно-упорядоченный список гиперссылок на справки об элементах языка BasicScript Language

Функция *GetScanTask(имя_задачи)* — при успешном завершении возвращает объект типа **ScanTask**. Если задача с заданным именем отсутствует или имя указано неверно, возвращается значение **Null**. Объект типа **ScanTask** является именем переменной типа **ScanTask**. Имя_задачи является символьной строкой, идентифицирующей задачу, которая входит в приложение. Имя задачи должно быть представлено прописными буквами и заключено в кавычки, например "TASK1".

Приведенный ранее пример 4.1 иллюстрирует также использование этой функции. См. также ScanTask (тип объекта), Start (метод).

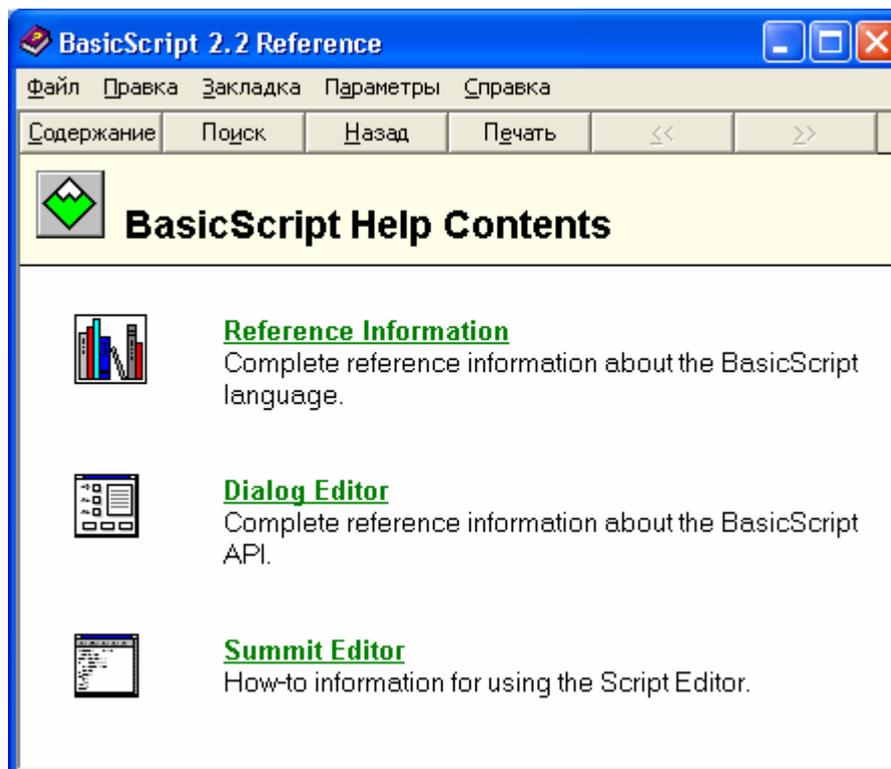


Рис. 4.80. Окно справки BasicScript Help Contents

Method Start, выполняемый для объекта типа **ScanTask** (задача), предназначен для включения задачи, для которой вызывается указанный метод, в список задач, вызываемых исполнительной средой в процессе выполнения стратегии. Метод может быть применен к объекту типа **ScanTask**. Указанными объектами являются задачи, исполняющиеся в рамках приложения. При вызове указанного метода производится немедленный запуск заданной задачи независимо от режима запуска, установленного в диалоговом окне **Scan Task Setup** редактора задач, появляющемся при выполнении команды **Setup | Task Properties...** (**Ctrl+T**). Возврат управления из данного метода происходит сразу после активизации задачи, для которой он вызывается.

Приведенный ранее пример 4.1 иллюстрирует использование и этого метода. См. также **ScanTask** (тип объекта), **GetScanTask** (функция).

Method Stop предназначен для исключения задачи, для которой вызывается указанный метод, из списка задач, вызываемых исполнительной средой в процессе выполнения приложения. Метод может быть применен к объекту типа **ScanTask**. Указанными объектами являются задачи, исполняющиеся в рамках приложения.

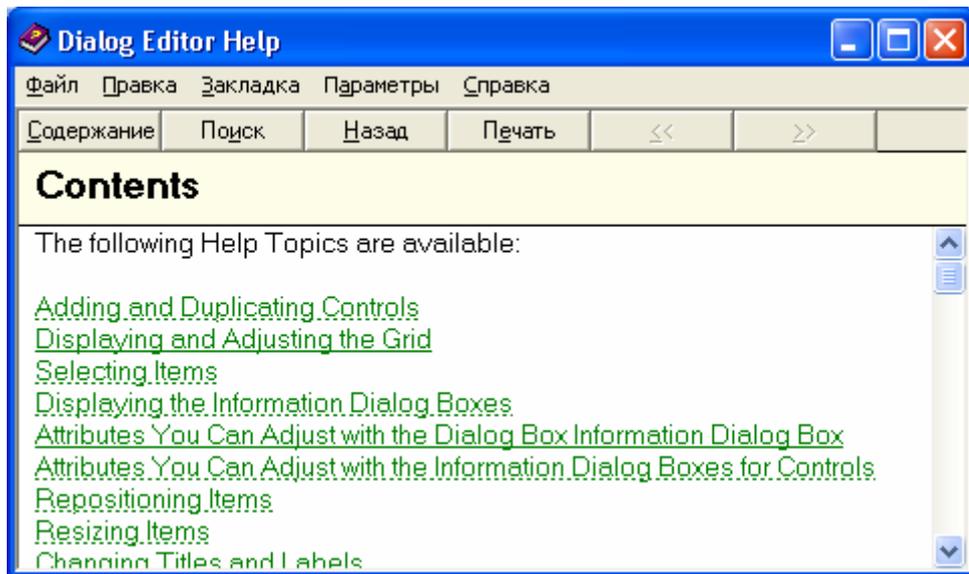


Рис. 4.81. Окно справки для создания диалогового окна

'Пример 4.2

'В данном примере останавливается задача TASK1.

' Предполагается, что эта задача уже запущена

Sub Main()

Dim MyTask As ScanTask

Dim IntFileLength As Integer

Set MyTask = GetScanTask("TASK1")

' Открываем файл для ввода и присваиваем ему номер 1

Open "test.dat" For Input As #1

' Определяем размер файла

IntFileLength = Lof(#1)

' Если файл слишком большой, то останавливаем задачу

If IntFileLength > 32000 Then

MyTask.Stop

End If

End Sub

См. также **ScanTask** (тип объекта), **GetScanTask** (функция).

Method SingleScan, выполняемый для объекта типа **ScanTask** (задача), предназначен для однократного исполнения задачи, для которой вызывается указанный метод. При вызове указанного метода производится немедленное однократное исполнение заданной задачи независимо от периода запуска, установленного в диалоговом окне **Scan Task Setup** редактора задач, появляющемся при выполнении команды **Setup — Task Properties...** (**Ctrl+T**). Возврат управления из данного метода происходит только по завершении однократного исполнения задачи, для которой он вызывается.

'Пример 4.3

'В данном примере 100 раз вызывается задача TASK1

```

Sub Main( )
    Dim IntI As Integer
    Dim MyTask As ScanTask
    Set MyTask = GetScanTask( "TASK1" )
    For IntI = 1 To 100
        MyTask.SingleScan
    Next IntI
End Sub

```

См. также **ScanTask** (тип), **GetScanTask** (функция).

Метод *GetStatus*, выполняемый для объекта типа **ScanTask** (задача), предназначен для получения текущего состояния задачи, для которой вызывается указанный метод. Результатом выполнения данного метода является возвращаемое целочисленное значение, отражающее текущее состояние проверяемой задачи. Значение 0 означает, что проверяемая задача не запущена и не остановлена. Значение 2 означает, что проверяемая задача находится в состоянии ожидания очередного вызова в соответствии с периодом сканирования, установленным в диалоговом окне **Scan Task Setup** редактора задач, появляющемся при выполнении команды **Setup | Task Properties... (Ctrl+T)**, или выполнение задачи приостановлено на заданный интервал времени. Значение 3 означает, что проверяемая задача находится в состоянии исполнения во время очередного вызова в соответствии с периодом сканирования, установленным в диалоговом окне **Setup | Task Properties... (Ctrl+T)**. Рекомендуется использовать данный метод до запуска или для остановки исполнения задачи.

' В примере 4.4 выполняется проверка состояния задачи TASK1

```

Sub Main( )
    Dim IntMyStatus As Integer
    Dim MyTask As ScanTask
    Set MyTask = GetScanTask( "TASK1" )
    MyStatus = MyTask.GetStatus
    If IntMyStatus = 0 Then
        MyTask.Start
        MyTask.Stop
    End If
End Sub

```

См. также **ScanTask** (тип), **GetScanTask** (функция), **Start** (метод), **Stop** (метод).

4.6.2.4.2. Работа с тегами

Средства работы с тегами предназначены для получения доступа к функциональным блокам и другим тегам, расположенным в центре обработки данных SCADA-системы. Они могут применяться в любой компоненте программного проекта (в основной процедуре, в предварительной и пост-процедурах задач, а также в Бейсик-процедурах). Понятие *тег* в определенном контексте может ассоциироваться как с идентификатором объекта (имя тега), так и с самим объектом определенного типа (класса). Примером *тега* является идентификатор функционального блока, входящего в задачу, однако имя тега записывается следующим образом. Для блока аналогового ввода, поле **Tag** диалогового окна

настройки параметров которого содержит идентификатор AI1, а поле **Description** — АЦП1, имя тега представляется как AI1:АЦП1.

Тип Tag используется для определения переменной, которая позволяет получить доступ к объекту, возвращаемому функцией GetTag().

```
' В примере 4.5 считывается значение на канале 0 блока
' аналогового ввода
```

```
Sub Main( )
    Dim MyTag As Tag
    Dim SngVoltage As Single
    Set MyTag = GetTag( "TASK2", "AI1" )
    SngVoltage = MyTag.Value
End Sub
```

См. также GetTag (функция), Value (свойство).

Функция *GetTag* при успешном выполнении возвращает объект типа **Tag** и имеет два аргумента. Первый аргумент является строкой и задает имя задачи, формы отображения или виртуальной задачи ("VIRTASK"), а второй также является строкой, содержащей имя функционального блока, элемента управления или виртуального тега (должно быть представлено прописными буквами).

Приведенный ранее пример 4.5 иллюстрирует использование и этой функции. См. также **Tag** (тип объекта), Value (свойство).

Свойство *Value* предназначено для получения доступа к значению, связанному с объектом типа **Tag**. Объект типа **Tag** в этом случае должен указывать на функциональный блок, имеющий один или несколько входов.

Приведенный ранее пример 4.5 иллюстрирует использование и этого свойства. См. также GetTag (функция), **Tag** (тип объекта).

Свойство *Array(номер_канала)* предназначено для получения доступа к значению, связанному с заданным каналом объекта типа **Tag**. Объект типа **Tag** в этом случае должен указывать на функциональный блок, имеющий один или несколько входов. Номер канала может принимать значения от 0 до 15.

```
' В примере 4.6 считывается значение на 16-ом канале блока
' аналогового ввода
```

```
Sub Main( )
    Dim MyTag As Tag
    Dim SngVoltage As Single
    Set MyTag = GetTag( "TASK2", "AI1" )
    SngVoltage = MyTag.Array( 15 )
End Sub
```

См. также GetTag (функция), **Tag** (тип объекта).

4.6.2.4.3. Управление окнами экранных форм

Функция *Display(номер_окна)* используется в основной, предварительной и пост-процедурах для переключения между несколькими окнами форм отображения, входящими в приложение. В качестве параметра оператору передается номер окна экранной формы. Например, для выдвигания на передний план окна экранной формы DISP1 следует использовать вызов функции Display(1).

```
' Пример 4.7 основного сценария стратегии
Sub Main( )
  Dim IntMyStatus As Integer
  Dim MyTask As ScanTask
  Set MyTask = GetScanTask( "TASK1" )
  IntMyStatus = MyTask.GetStatus
  If IntMyStatus = 3 Then
    ' Отображение окна DISP1 на переднем плане
    Display( 1 )
  End If
End Sub
```

4.6.2.4.4. Работа с Бейсик-процедурами

Средства работы с Бейсик-процедурами предназначены для управления выходными каналами функциональных блоков Бейсик-сценарий. Указанные средства могут применяться *только в Бейсик-процедурах*. Далее приводится описание операторов, выполняющих управление выходными каналами функциональных блоков Бейсик-сценарий.

Оператор Outputf используется для передачи одного вещественного значения из блока Бейсик-сценария другому функциональному блоку приложения и имеет два формата:

```
Outputf значение
Outputf канал, значение
```

Здесь канал — целое число в диапазоне от 0 до 7, представляющее номер канала, по которому будет осуществляться передача вещественного значения, а значение — имя переменной типа **Single** или значение, передаваемое другому функциональному блоку приложения. Если явно не указан номер канала, по которому выводится значение с плавающей точкой, предполагается канал с номером 0.

```
' В примере 4.8 выполняется вывод вещественного значения по
' каналу 0
  Dim MyTag As Tag
  Dim SngVoltage As Single
  Set MyTag = GetTag( "TASK2", "A11" )
  SngVoltage = MyTag.Value
  If SngVoltage > 0.0 Then
    Outputf SngVoltage
  Else
    Outputf -SngVoltage
  End If
```

См. также *Outputi* (оператор), *Outputl* (оператор), *Outputs* (оператор)

Оператор Outputi используется для передачи одного целочисленного значения из блока Бейсик-сценария другому функциональному блоку приложения и имеет два формата:

```
Outputi значение
Outputi канал, значение
```

Здесь канал — целое число в диапазоне от 0 до 7, представляющее номер канала, по которому будет осуществляться передача целочисленного значения, а значение —

имя переменной типа **Integer** или целочисленное значение, передаваемое другому функциональному блоку приложения. Если явно не указан номер канала, по которому выводится значение с плавающей точкой, предполагается канал с номером 0.

```
' В примере 4.9 выполняется вывод целочисленного значения по
' каналу 0
  Dim MyTag As Tag
  Dim IntRamp As Integer
  Set MyTag = GetTag( "TASK2", "RMP1" )
  IntRamp = MyTag.Value
  If IntRamp > 3 Then
    Outputi IntRamp
  Else
    Outputi -1
  End If
```

См. также Outputf (оператор), Outputl (оператор), Outputs (оператор)

Оператор Outputl используется для передачи одного целочисленного значения двойной точности из блока Бейсик-сценария другому функциональному блоку приложения и имеет два формата:

```
Outputl значение
Outputl канал, значение
```

Здесь канал — целое число в диапазоне от 0 до 7, представляющее номер канала, по которому будет осуществляться передача целочисленного значения двойной точности, а значение — имя переменной типа **Long** или целочисленное значение двойной точности, передаваемое другому функциональному блоку приложения. Если явно не указан номер канала, по которому выводится значение с плавающей точкой, предполагается канал с номером 0.

```
' В примере 4.10 выполняется вывод значения целого
' типа двойной точности по двум каналам
  Dim LngCount As Long
  ' ...
  If LngCount >= 0 Then
    Outputl 0, LngCount
  Else
    Outputl 1, LngCount
  End If
```

См. также Outputf (оператор), Outputi (оператор), Outputs (оператор).

Оператор Outputs используется для передачи одной строки из блока Бейсик-сценария другому функциональному блоку приложения и имеет два формата:

```
Outputl строка$
Outputl канал, строка$
```

Здесь канал — целое число в диапазоне от 0 до 7, представляющее номер канала, по которому будет осуществляться передача строки, а значение — имя переменной типа **String** или строковое значение, передаваемое другому функциональному блоку приложения. Если явно не указан номер канала, по которому выводится значение с плавающей точкой, предполагается канал с номером 0.

```
' В примере 10 выполняется вывод строки
Dim StrS1 As String
Dim StrS2 As String
StrS1 = "Hello,"
StrS2 = "World!"
Outputs 3, StrS1 + StrS2
```

См. также Outputf (оператор), Outputi (оператор), Outputl (оператор).

4.6.2.5. Использование в VBA-программе диалоговых окон.

Управляющие элементы

Проиллюстрируем создание и использование окна диалога на примере диалогового окна с кнопками **OK**, **Cancel**, управляющими элементами отмечаемая кнопка (**Check box**), элемент группирования (**Group box**), статический текст (**Text**), элемент редактирования (**Text box**) и раскрывающийся список (**Drop list box**).

Для создания демонстрационного примера выполните следующие действия.

1. *Первое действие.* Запустите построитель приложений двойным щелчком левой кнопкой мыши на ярлыке GeniDAQ Builder 4.25, расположенном на рабочем столе. В результате на экран будет выведено главное окно построителя приложений (см. рис. 4.6). Для запуска построителя задач в демонстрационном режиме нажмите кнопки **Exit** и **OK**. Демонстрационный режим отличается непрерывным двухчасовым интервалом работы и меньшим количеством входных и выходных каналов. В нашем случае это несущественно. Нажмите кнопку **New** на панели инструментов (ее легко найти, так как все кнопки снабжены всплывающими подсказками). После этого в главном окне построителя приложений появится диалоговое окно настройки, которое может быть настроено так, как это показано на рис. 4.82. После нажатия кнопки **OK** и подтверждения необходимости создания каталога DemoDialog, который до этого не существовал, в главном окне построителя задач появятся пустые окна редактора задач (**TASK1**) и редактора форм отображения (**DISP1**). Произведем щелчок мышью в области окна редактора задач и окно с заголовком **Task Designer: TASK1** будет выдвинуто на передний план. В окне построителя задач появится еще одна панель инструментов, содержащая пиктограммы встроенных функциональных блоков редактора задач.

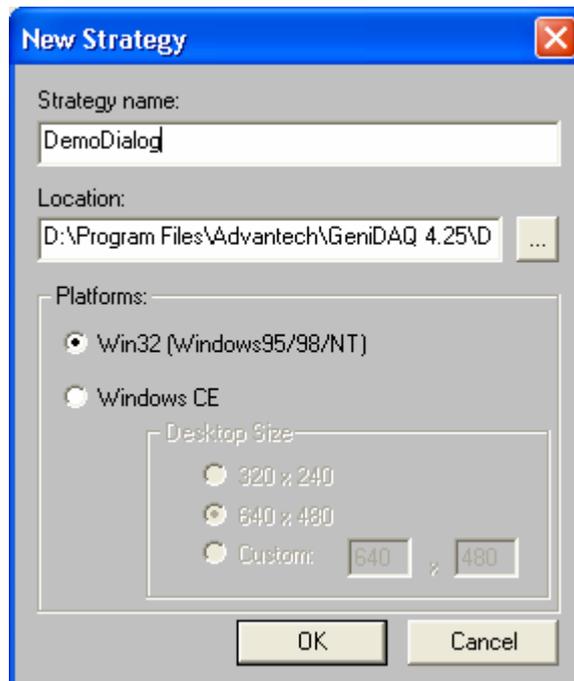


Рис. 4.82. Конфигурирование приложения **DemoDialog**

2. *Второе действие.* Задайте временные параметры запуска задачи. Для этого нажмите кнопку **Task Properties** на панели инструментов или выполните команду **Setup | Task Properties**, в появившемся окне задайте требуемое значение периода (см. рис. 4.8) и нажмите кнопку **OK**. В нашем примере задан период запуска приложения 100 миллисекунд.
3. *Третье действие.* На панели инструментов выберите кнопку с мнемоникой **BS** (BASIC Script), поместите на нее курсор мыши, щелкните левой кнопкой мыши, переместите мышь в окно редактора задач и еще раз щелкните левой кнопкой мыши. При этом в окне редактора задач появится функциональный блок Бейсик-сценария с позиционным обозначением **SCR1** (рис. 4.83).
4. *Четвертое действие.* Теперь сделайте активным окно **DISP1** редактора форм отображения, для чего щелкните по этому окну левой кнопкой мыши. При этом в верхней части главного окна появится панель инструментов редактора форм отображения. На панели инструментов выберите кнопку **Binary Button** (нажимаемая кнопка) и с помощью мыши перетащите ее и положите во внутреннюю область окна редактора форм отображения (рис. 4.84). Для настройки свойств кнопки щелкните дважды левой кнопкой мыши над пиктограммой кнопки. В результате появится диалоговое окно настройки. Выполните настройку кнопки в соответствии с рис. 4.85, а затем с помощью

кнопки **OK** закройте окно (в ответ на предупреждение еще раз нажмите кнопку **OK**). С помощью этой кнопки будем управлять окном диалога.

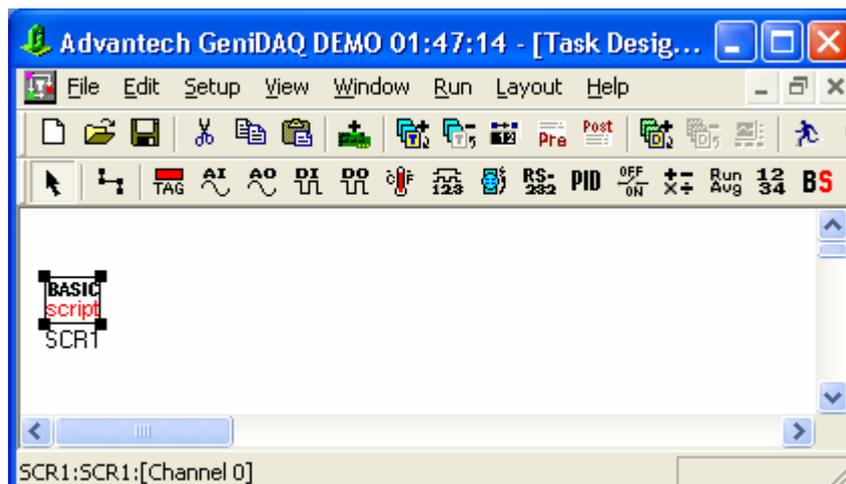


Рис. 4.83. Вид окна редактора задач приложения DemoDialog

5. *Пятое действие.* Теперь сделайте вновь активным окно **TASK1** редактора задач, для чего щелкните по этому окну левой кнопкой мыши. При этом в верхней части главного окна появится панель инструментов редактора задач. На панели инструментов выберите кнопку **Tag** (tag) и с помощью мыши перетащите ее и положите во внутреннюю область окна редактора форм отображения (рис. 4.86). Для настройки свойств тега щелкните дважды левой кнопкой мыши над пиктограммой тега. В результате появится диалоговое окно настройки. Выполните настройку тега в соответствии с рис. 4.87, а затем с помощью кнопки **OK** закройте окно. Созданный тег будет использован для передачи в Бейсик-процедуру информации о состоянии кнопки в окне отображения.
6. *Шестое действие.* Двойным щелчком левой кнопки мыши по функциональному блоку Бейсик-сценария активируйте ИСП Бейсик-процедуры и выполните команду **Edit | Insert New Dialog...** В результате появится ИСП диалогового окна, представленная на рис. 4.88. Разместите в шаблоне окна диалога управляющие элементы группирования, статического текста, редактирования, отмечаемую кнопку и раскрывающийся список в соответствии с рис. 4.89. Настройте диалоговое окно и добавленные управляющие элементы в соответствии с рис. 4.90-4.95. Командой **File | Save As...** сохраните созданный диалог в файле на магнитном диске, а командой **File | Exit And Return** включите шаблон диалога в Бейсик-процедуру.
7. *Седьмое действие.* Выполните программную поддержку созданного диалога. С этой целью в Бейсик-процедуру введите следующий текст:

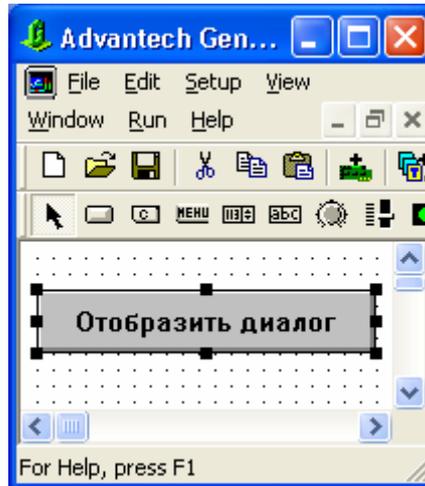


Рис. 4.84. Окно редактора форм отображения приложения DemoDialog

```
' Создание и обработка диалога
' Массив для раскрывающегося списка
Public DropListBox1$( 2 )
' Шаблон диалога (создается автоматически)
Begin Dialog UserDialog , ,264,113,"TestDialog"
    OKButton 220,8,40,14
    CancelButton 220,28,40,14
    GroupBox 4,4,208,104,"Блок группирования",.GroupBox1
    TextBox 8,16,44,12,.TextBox1
    Text 60,17,108,8,"Элемент редактирования",.Text1
    CheckBox 8,36,160,8,"Отмечаемая кнопка",.CheckBox1
    DropListBox 8,50,68,58,DropListBox1$,.DropListBox1
    Text 84,53,124,8,"Раскрывающийся список",.Text2
End Dialog
' Инициализация, создание и обработка диалога
Sub SCR1()
    ' Тег кнопки, управляющей диалогом
    Dim button As Tag
    ' Получение значения состояния кнопки
    Set button = GetTag( "DISP1", "BBTN1" )
    If button.value = 1 Then
        ' Кнопка нажата - инициализация, создание и обработка
        ' диалога
        ' Инициализация
        ' Данные для раскрывающегося списка
        DropListBox1$( 0 ) = "Один"
        DropListBox1$( 1 ) = "Два"
        DropListBox1$( 2 ) = "Три"
        Dim dlg As UserDialog ' Объект диалога
        ' Индекс элемента раскрывающегося списка
        dlg.DropListBox1 = 2
    End If
End Sub
```

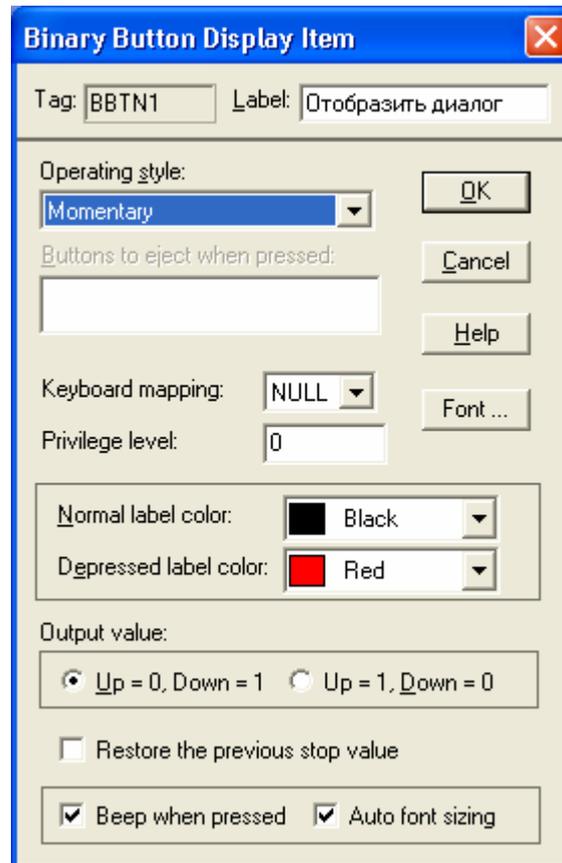


Рис. 4.85. Конфигурирование кнопки

```
' Текст для элемента редактирования
dlg.TextBox1 = "Пример"
' Отмечаемая кнопка выбрана
dlg.CheckBox1 = 1
' Для возвращаемого элемента диалога (-1 при нажатии кнопки
' ОК)
Dim IntrC As Integer
' Создание диалога и работа с ним
IntrC = Dialog( dlg )
' Обработка диалога
If IntrC = -1 Then
    ' Нажата кнопка ОК
    ' Состояние раскрывающегося списка
    MsgBox "Раскрывающийся список: " & _
        DropListBox1$( dlg.DropListBox1 )
    ' Состояние элемента редактирования
    MsgBox "Элемент редактирования: " & dlg.TextBox1
```

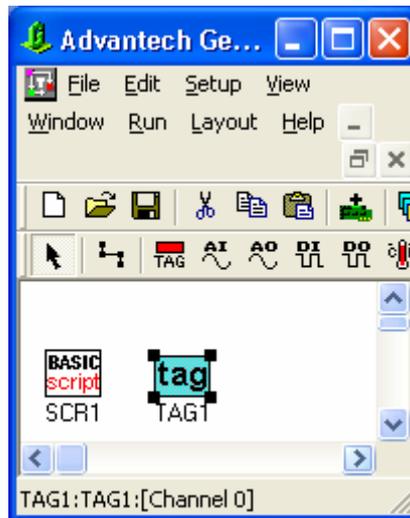


Рис. 4.86. Окно редактора задач приложения DemoDialog

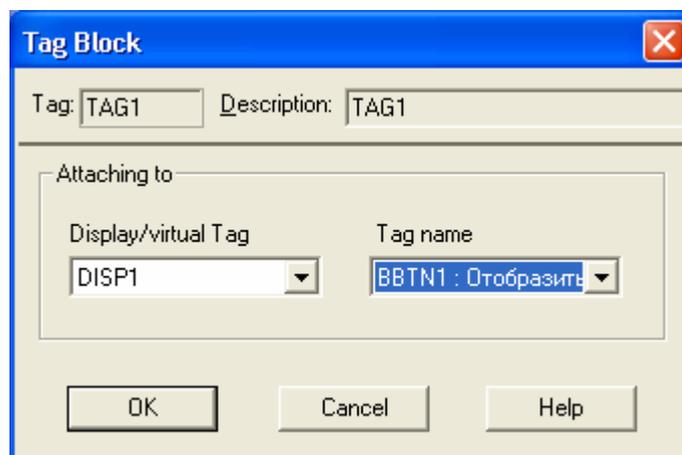


Рис. 4.87. Конфигурирование тега

```

' Состояние отмечаемой кнопки
MsgBox "Отмечаемая кнопка: " & dlg.CheckBox1
End If
End Sub
End Sub

```

Командой **File | Close** закройте ИСР программы и сохраните Бейсик-процедуру.

8. *Восьмое действие.* Протестируйте созданное приложение. С этой целью кнопкой **Start** активизируйте приложение и в окне отображения нажмите кнопку **Отобразить диалог**. В результате появится диалоговое окно, представленное на

рис. 4.96. В этом окне изменим текст в элементе редактирования, состояние отмечаемой кнопки, выберем другой элемент в раскрывающемся списке и для считывания информации из окна диалога нажмем кнопку **ОК**. В результате появится последовательность сообщений, отображающих полученную информацию (рис. 4.97 — 4.99).

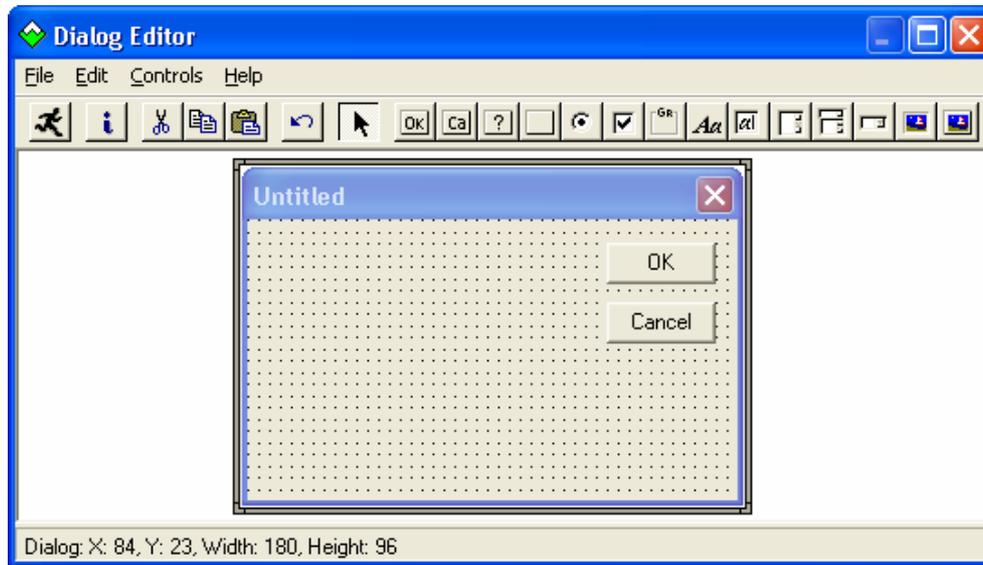


Рис. 4.88. ИСР диалогового окна

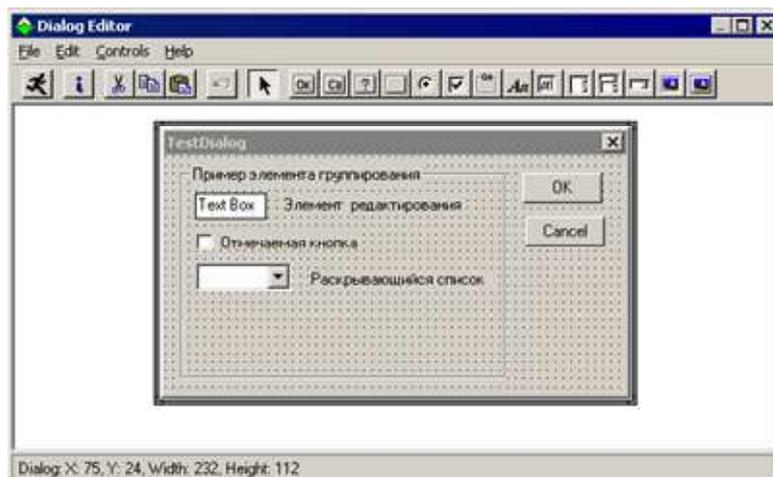


Рис. 4.89. Вид окна диалога



Рис. 4.90. Конфигурирование окна диалога

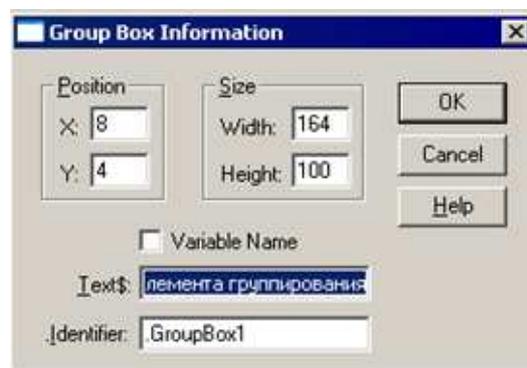


Рис. 4.91. Конфигурирование элемента группирования

9. *Девятое действие.* Завершите работу приложения. С этой целью кнопкой **Stop** остановите работу приложения и командой **File | Exit** завершите работу SCADA-системы.

4.6.3. Упражнения

Упражнение 4.17. Повтор создания и тестирования приложения DemoDialog

Повторите создание и тестирование рассмотренного ранее приложения **DemoDialog**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

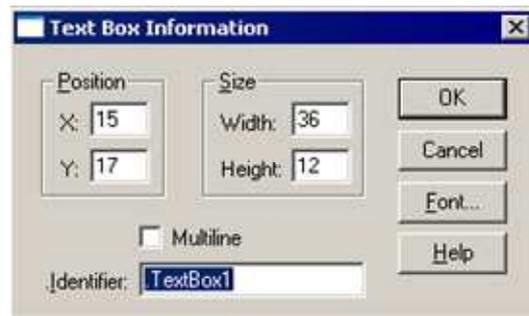


Рис. 4.92. Конфигурирование элемента редактирования

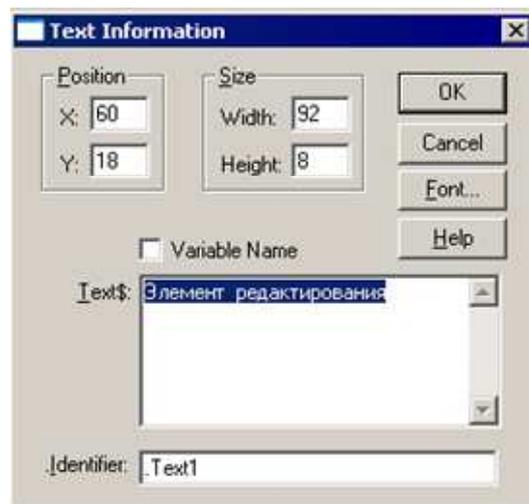


Рис. 4.93. Конфигурирование статического текста

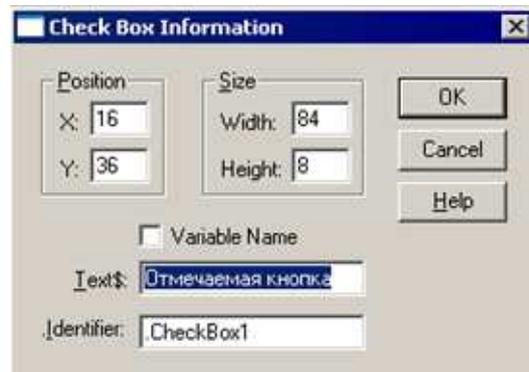


Рис. 4.94. Конфигурирование отмечаемой кнопки



Рис. 4.95. Конфигурирование раскрывающегося списка

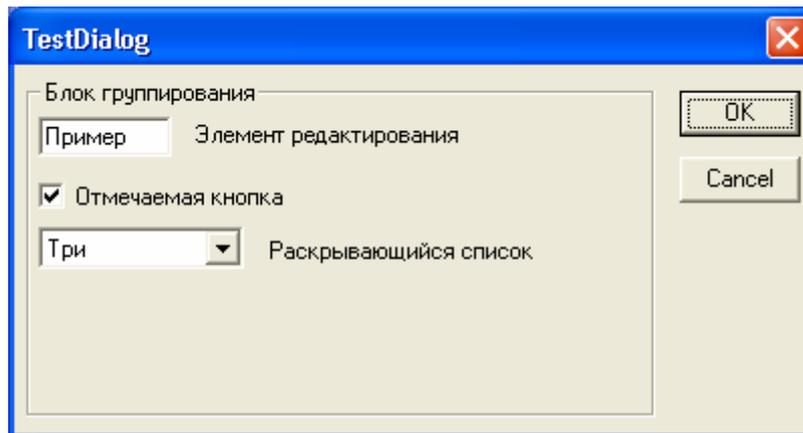


Рис. 4.96. Вид окна диалога

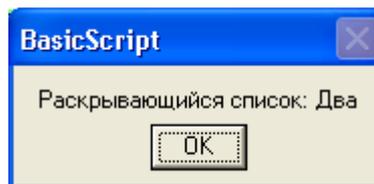


Рис. 4.97. Информация, полученная из раскрывающегося списка

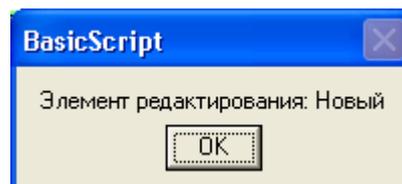


Рис. 4.98. Информация, полученная из элемента редактирования

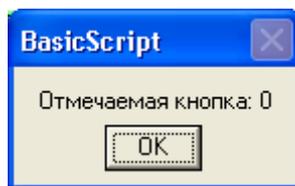


Рис. 4.99. Информация, полученная из отмечаемой кнопки

4.7. Занятие 6 "Использование функционального блока Бейсик-сценария"

Цель занятия состоит в формировании у пользователя представления о правилах использования функционального блока *Бейсик-сценария (BASIC Script)* редактора задач. Используются функциональные блок *тег (Tag)*, блок *аналогового ввода (Analog Input)*, блок *Бейсик-сценария (BASIC Script)* редактора задач и инструменты *инкрементный регулятор (Numeric Control)*, *индикатор (Indicator)* и *цифровой индикатор (Numeric String)* редактора форм отображения. Блок *Бейсик-сценария* используется на занятии следующим образом. С его помощью получается значение аналогового сигнала канала 0 программного эмулятора сигналов **Advantech DEMO I/O**. Полученное значение сравнивается с величиной, вводимой пользователем с помощью *инкрементного регулятора*. Если значение сигнала на канале 0 превышает вводимое пользователем, то цвет *индикатора* меняется с зеленого на красный. В ином случае — цвет изменяется на зеленый.

4.7.1. Используемый инструментарий

Новым инструментом на данном занятии является *функциональный блок Бейсик-сценария (BASIC Script)* редактора задач. Блок разработан для обеспечения максимальной гибкости программирования в GeniDAQ, позволяющей реализовывать эффективные вычисления, логические операции, условные переходы, ветвления и циклы. Как указывалось ранее, в блоке используется язык программирования, синтаксис и функции которого совместимы с принятыми в Microsoft Visual Basic и Microsoft Visual Basic for Applications (VBA). Однако язык программирования Бейсик-сценариев является интерпретатором, что вызвано необходимостью достижения компромисса между простотой в применении и быстродействием. Блок Бейсик-сценария предназначен для реализации небольших алгоритмов обработки данных. Разработка и использование больших и сложных программ не рекомендуется. Выходы блока (их всего 8) блока могут соединяться с неограниченным количеством других функциональных блоков стратегии. Имеется возможность создания программ, выполняющих обработку значений на входах блока Бейсик-сценария и вывод вычисленных значений на его выходы, а также позволяющих пропускать циклы стратегии, при которых не происходит вывод значений переменных другим функциональным блокам. В последнем случае, все блоки стратегии, присоединенные к выходам блока с подобным алгоритмом работы, исключаются из обработки в пропускаемых циклах. После двойного щелчка левой

кнопкой мыши на пиктограмме блока Бейсик-сценария появляется окно редактора Бейсик-сценариев (рис. 4.77), подробно описанного в предыдущем разделе. С помощью данного редактора имеется возможность отладки, трассировки и редактирования программного кода сценариев. Таким образом, редактор сценариев представляет мощное средство разработки процедур обработки данных, специфичных для задачи пользователя. Окно редактора сценариев является модальным, что требует его закрытия перед продолжением работы с другими редакторами GeniDAQ. Блок Бейсик-сценария имеет доступ к данным, расположенным в центре обработки данных GeniDAQ. Таким образом, входные связи, присоединяемые к блоку, не несут никакой смысловой нагрузки и игнорируются.

4.7.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.101.
2. Выполните *второе действие* из занятия 1. В нашем примере также задан период запуска приложения 100 миллисекунд.
3. *Третье действие.* Добавьте в окно отображения **DISP1** цифровой индикатор, индикатор, инкрементный регулятор и две текстовых строки (рис. 4.102).
4. *Четвертое действие.* Выдвиньте на передний план окно редактора задач **TASK1** и разместите в нем блок аналогового входа **AI1**, блок тег и блок Бейсик-сценария. Выполните настройку блока аналогового ввода на прием измерительной информации от канала 0 программного эмулятора **Advantech DEMO I/O**. С помощью проводника установите связь между блоком аналогового ввода **AI1** и блоком Бейсик-сценария **SCR1**. Окно редактора задач приобретет вид, показанный на рис. 4.103. Перед соединением функциональных блоков тег и **SCR1** настройте тег в соответствии с рис. 4.104 и соедините блоки с помощью проводника. При этом окно редактора задач приобретет окончательный вид, показанный на рис. 4.105. Сконфигурируйте блок Бейсик-сценария в соответствии с рис. 4.100.
5. *Пятое действие.* Перейдите в окно редактора форм отображения и настройте цифровой индикатор, индикатор и инкрементный регулятор в соответствии с рис. 4.106-4.108.

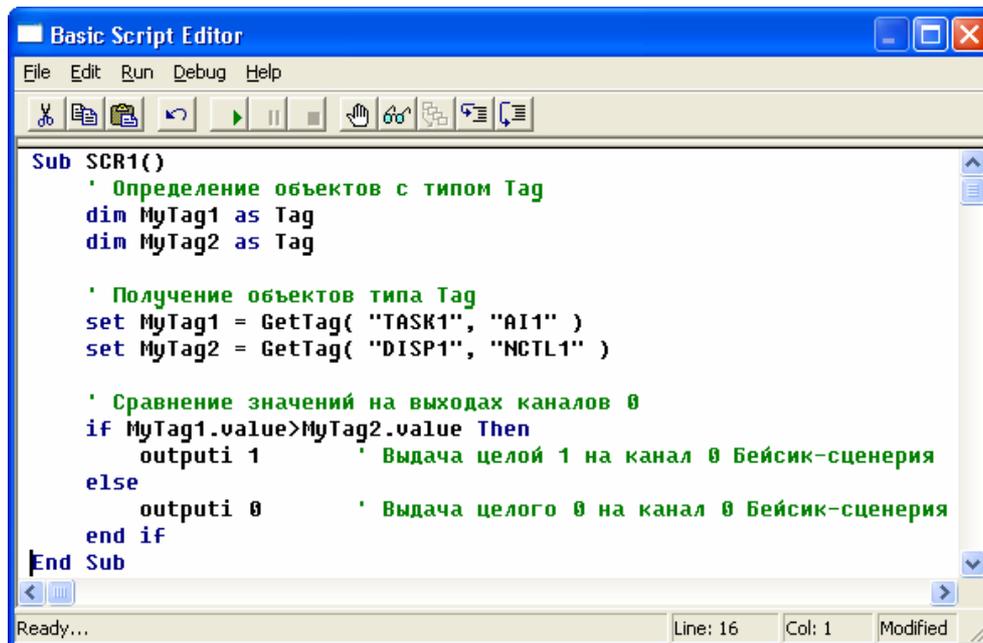


Рис. 4.100. Конфигурирование функционального блока **Бейсик-сценария**

6. *Шестое действие.* Сохраните созданный проект (стратегию). Для этого выполните команду **File | Save As**, в появившемся диалоговом окне укажите необходимую информацию и нажмите кнопку **Сохранить**. Это нужно выполнить для обоих файлов проекта с расширениями **.gni** и **.HLD**. После сохранения файлов созданного проекта запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.109). В процессе работы приложения обратите внимание на состояние индикатора и проанализируйте, почему индикатор ведет себя таким образом. Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
7. *Седьмое действие. Девятое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

4.7.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.18. Повтор создания и тестирования приложения "Занятие 6"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 6**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

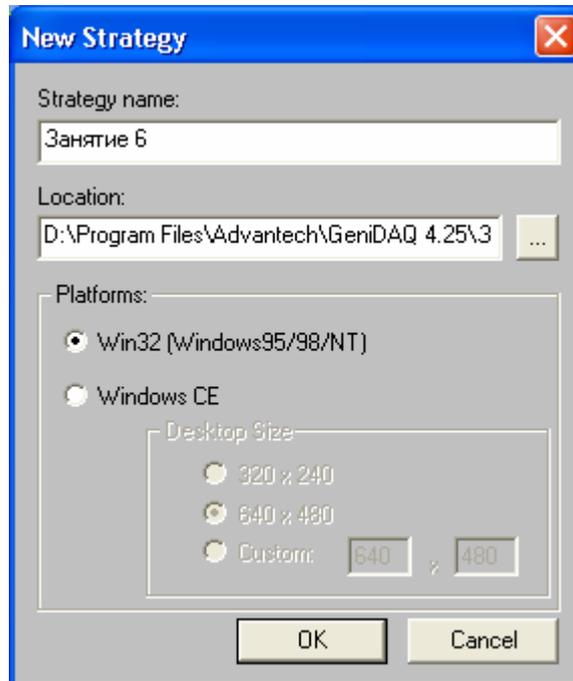


Рис. 4.101. Диалоговое окно конфигурирования приложения

Упражнение 4.19

Разработайте стратегию, обладающую следующими возможностями. В окне редактора задач поместите функциональный блок аналогового ввода, использующий эмулятор пилообразного сигнала, и функциональный блок Бейсик-сценария. В окне редактора форм отображения поместите два цифровых индикатора. Настройте первый из цифровых индикатора таким образом, чтобы он отображал сигнал на выходе блока аналогового ввода. Запрограммируйте блок Бейсик-сценария так, чтобы настройка обеспечила отображение на втором индикаторе максимального значения на выходе блока аналогового ввода.

Упражнение 4.20

Разработайте стратегию, обладающую следующими возможностями. В окне редактора задач поместите функциональный блок аналогового ввода, использующий эмулятор гармонического сигнала, и функциональный блок Бейсик-сценария.

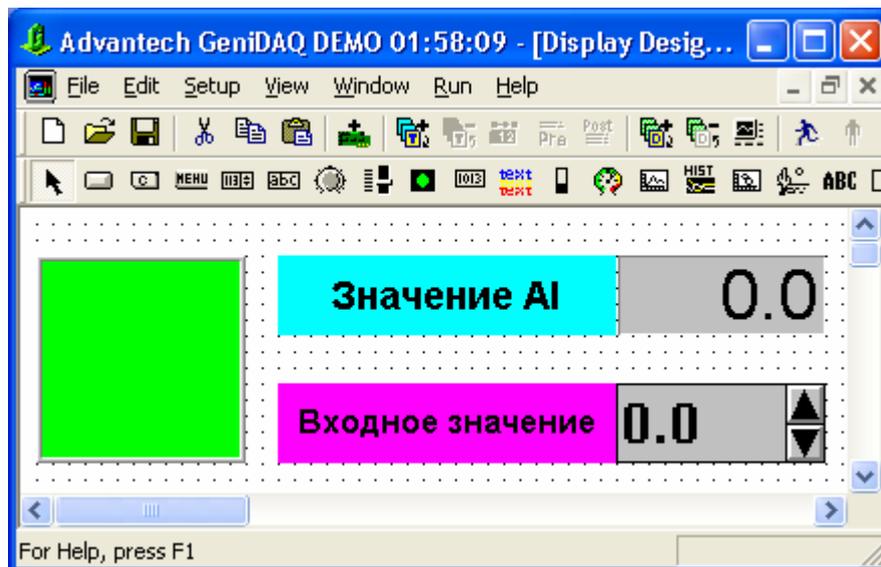


Рис. 4.102. Вид окна редактора форм отображения после добавления управляющих элементов

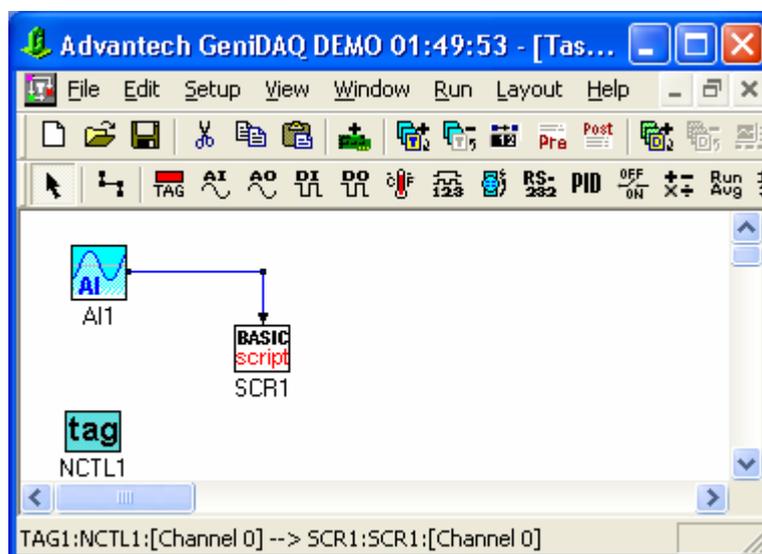


Рис. 4.103. Вид окна редактора задач

В окно редактора форм отображения поместите два цифровых индикатора. Настройте первый из цифровых индикатора таким образом, чтобы он отображал сигнал на выходе блока аналогового ввода. Запрограммируйте блок Бейсик-сценария

так, чтобы настройка обеспечила отображение на втором индикаторе минимального значения на выходе блока аналогового ввода.

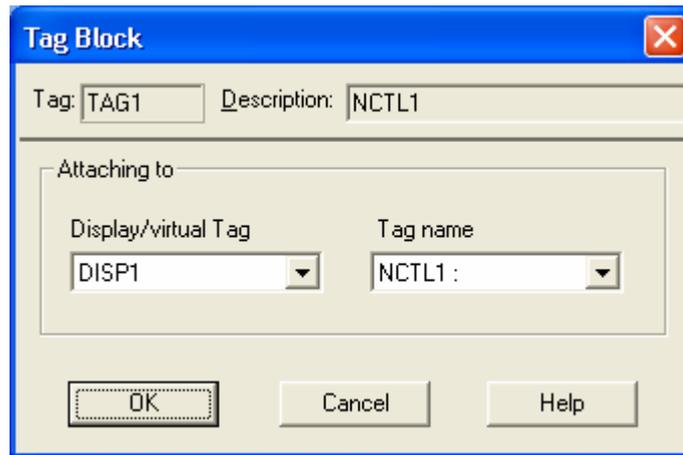


Рис. 4.104. Конфигурирование блока *me*

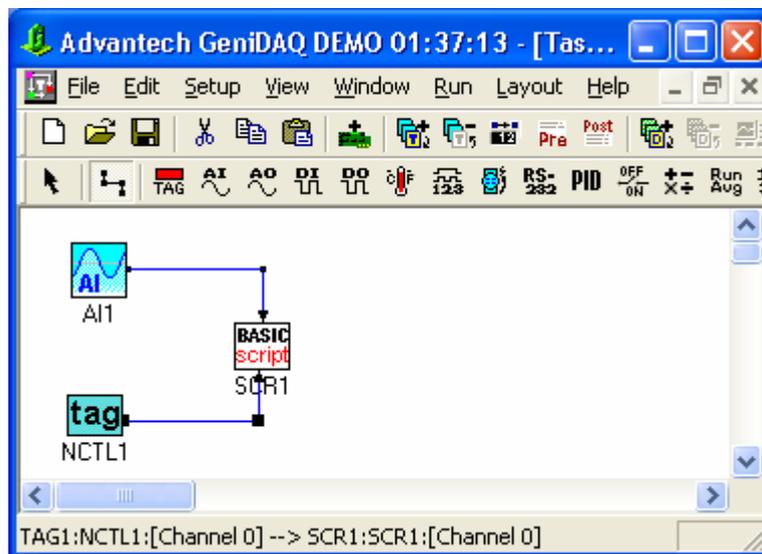


Рис. 4.105. Окончательный вид окна редактора задач

Упражнение 4.21

Разработайте стратегию, обладающую следующими возможностями. В окно редактора задач поместите функциональный блок аналогового ввода, использующий эмулятор случайного сигнала, и функциональный блок Бейсик-сценария. В окно редактора форм отображения поместите два цифровых индикатора. Настройте

первый из цифровых индикатора таким образом, чтобы он отображал сигнал на выходе блока аналогового ввода. Запрограммируйте блок Бейсик-сценария так, чтобы настройка обеспечила отображение на втором индикаторе максимального значения на выходе блока аналогового ввода.

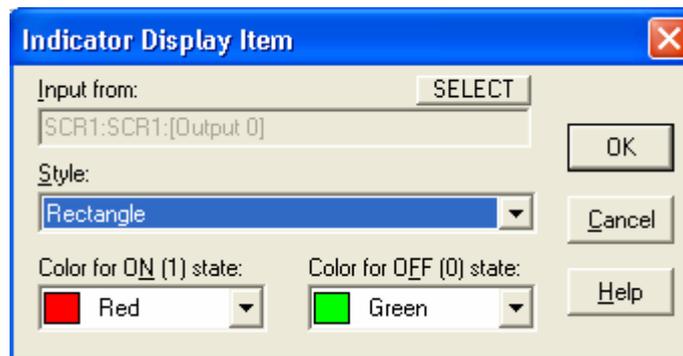


Рис. 4.106. Конфигурирование индикатора

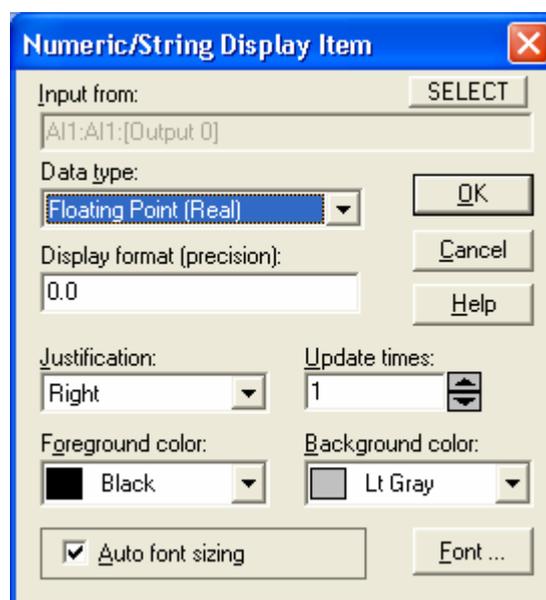


Рис. 4.107. Конфигурирование цифрового индикатора

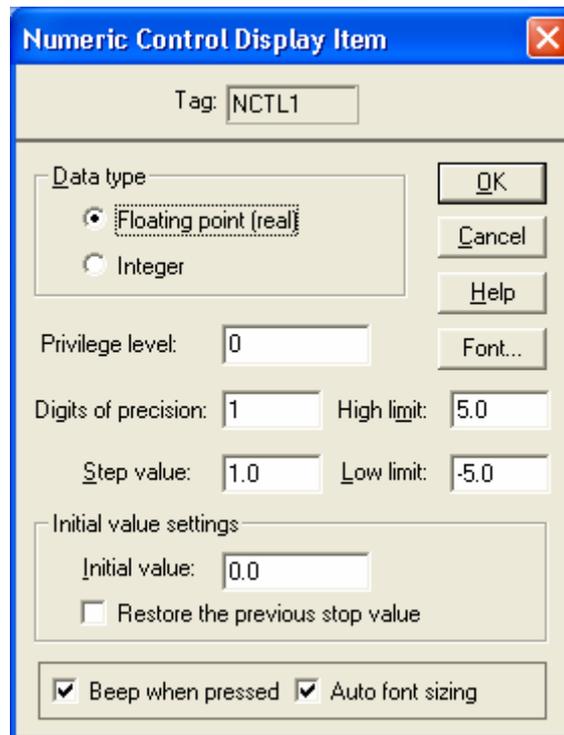


Рис. 4.108. Конфигурирование инкрементного регулятора

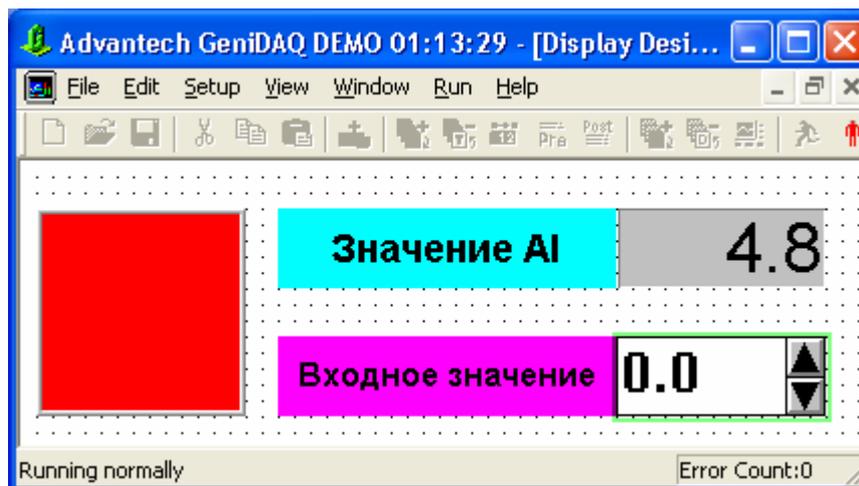


Рис. 4.109. Демонстрация работы проекта в окне отображения

4.8. Занятие 7 "Совместное использование функциональных блоков Бейсик-сценария и виртуального тега"

Цель занятия состоит в изучении правил совместного применения функциональных блоков Бейсик-сценария (**BASIC Script**) и виртуального тега (**Virtual Tag**). Используются функциональный блок тег (**Tag**), который связывается с виртуальным тегом, блок аналогового ввода (**Analog Input**), блок Бейсик-сценария редактора задач и инструменты индикатор (**Indicator**), два цифровых индикатора (**Numeric String**) и две текстовых строки (**Text String**) редактора форм отображения. Блок Бейсик-сценария используется на занятии следующим образом. С его помощью получается значение аналогового сигнала канала 0 программного эмулятора сигналов **Advantech DEMO I/O**. Полученное значение сравнивается с величиной, хранимой в виртуальном теге. Если значение сигнала на канале 0 превышает хранимое в виртуальном теге, то это значение запоминается в виртуальном теге и на выходе блока Бейсик-сценарий формируется единичный сигнал. Иначе — значение, хранимое в виртуальном теге, не изменяется, а на выходе блока Бейсик-сценарий формируется нулевой сигнал. Цвет индикатора — красный, пока значение сигнала на канале 0 превышает хранимое в виртуальном теге. В противоположном случае — цвет изменяется на зеленый.

4.8.1. Используемый инструментарий

Новым инструментом на данном занятии является *функциональный блок виртуальный тег (Virtual Tag)*. Виртуальный тег, созданный в редакторе задач, сохраняется в центре обработки данных так же, как и остальные встроенные функциональные блоки. Виртуальные теги доступны всем задачам, что обеспечивает возможность их использования для обмена данными между несколькими задачами.

Для создания виртуального тега выполните команду **Setup | Add/Delete Virtual Tags** и на экран будет выведено диалоговое окно, содержащее таблицу виртуальных тегов (рис. 4.110). Для создания виртуального тега и добавления его в таблицу следует нажать кнопку **Add Tag**, в появившемся окне диалога (рис. 4.111) ввести его имя, тип, начальное значение и нажать кнопку **OK**. При этом виртуальный тег с введенным именем будет создан и добавлен в таблицу виртуальных тегов (см. приведенный ранее рис. 4.110). Для изменения параметров виртуального тега надо выбрать его в списке тегов (см. приведенный ранее рис. 4.110), нажать кнопку **Update**, в появившемся окне диалоговом окне (см. приведенный ранее рис. 4.111) задать требуемые параметры и нажать кнопку **OK**. Для удаления виртуального тега следует выбрать его имя в таблице виртуальных тегов и нажать кнопку **Delete**. После модификации таблицы виртуальных тегов нажатием кнопки **Close** можно закрыть диалоговое окно, показанное на рис. 4.110.

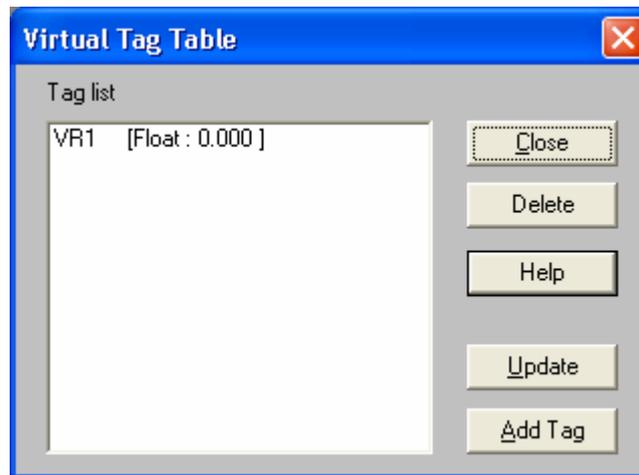


Рис. 4.110. Диалоговое окно для модификации таблицы виртуальных тегов

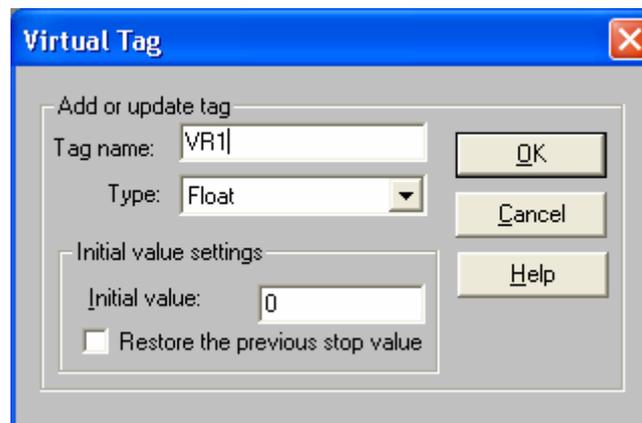


Рис. 4.111. Задание параметров добавляемого виртуального тега

После создания, виртуальные теги не помещаются в набор инструментов редактора задач. Доступ к виртуальным тегам в редакторе задач может быть получен через блок тег. Для этого следует в окне конфигурирования тега (рис. 4.112) в списке **Display/virtual Tag** выбрать значение **VIRTASK**, а в списке **Tag name** выбрать имя виртуального тега (например, **VR1**), к которому требуется получить доступ. После нажатия кнопки **OK**, блок тег может быть соединен с другими функциональными блоками стратегии, имеющими входы и выходы. Таким образом, виртуальный тег может быть как источником, так и приемником данных. Например, ему могут передаваться значения от блока аналогового ввода или он может передавать данные блоку дискретного вывода.

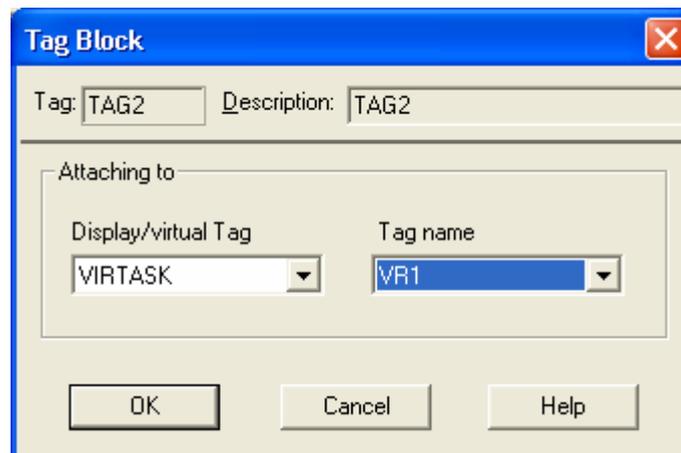


Рис. 4.112. Диалоговое окно для настройки доступа к виртуальному тегу

Значение, связанное с виртуальным тегом, может быть отображено на экране монитора путем установления связи с соответствующим элементом отображения (рис. 4.113).

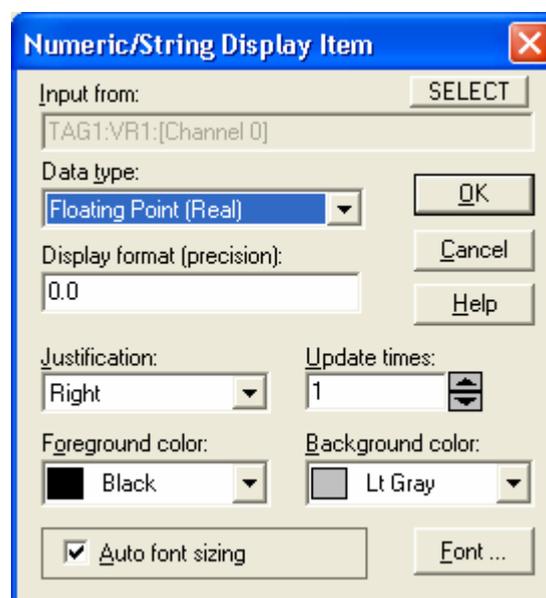
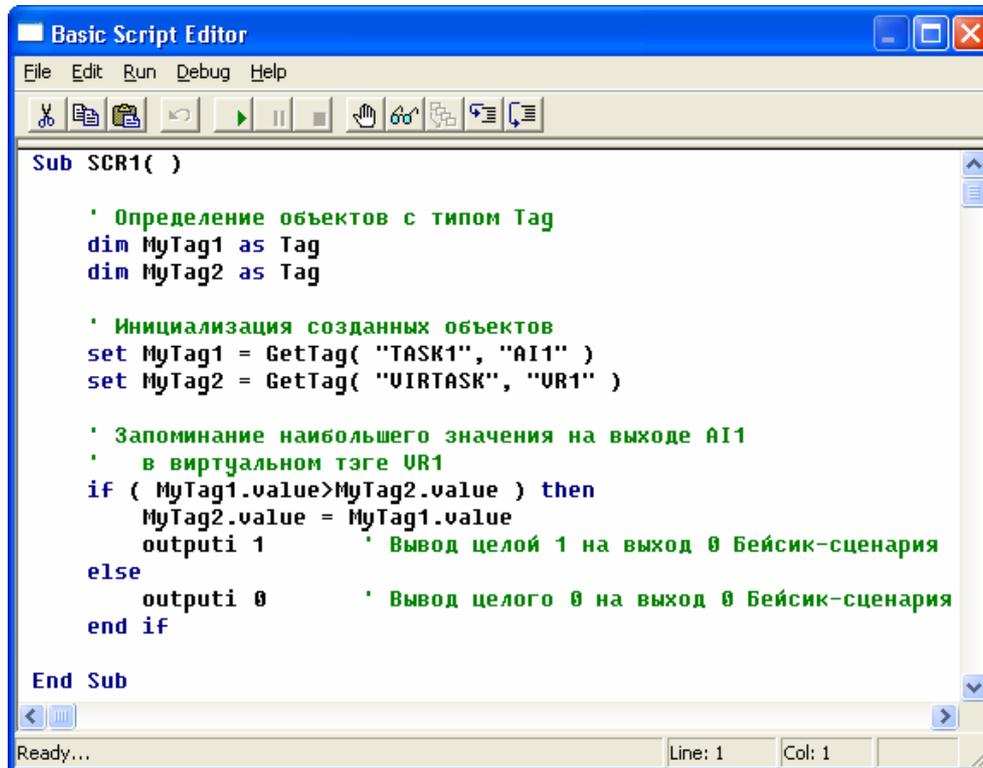


Рис. 4.113. Связь виртуального тега с элементом отображения редактора форм отображения

В Бейсик-сценарии виртуальный тег является тем же объектом, что и другие функциональные блоки стратегии. Имеется возможность получить и установить значение, связанное с виртуальным тегом, путем использования функций `GetTag()`

и SetTag(). Сказанное иллюстрирует рис. 4.114. Отличие между виртуальным тегом и другими функциональными блоками редактора задач состоит в том, что после установки новых значений предыдущие значения, связанные с блоками редактора задач, не могут быть вновь считаны задачами стратегии из центра обработки данных. Таким образом, использование виртуальных тегов дает возможность передачи данных из Бейсик-сценариев и других приложений задачам и экранным формам текущей стратегии.



```

Sub SCR1( )

  ' Определение объектов с типом Tag
  dim MyTag1 as Tag
  dim MyTag2 as Tag

  ' Инициализация созданных объектов
  set MyTag1 = GetTag( "TASK1", "AI1" )
  set MyTag2 = GetTag( "VIRTASK", "UR1" )

  ' Запоминание наибольшего значения на выходе AI1
  ' в виртуальном тэге UR1
  if ( MyTag1.value > MyTag2.value ) then
    MyTag2.value = MyTag1.value
    outputi 1      ' Вывод целой 1 на выход 0 Бейсик-сценария
  else
    outputi 0      ' Вывод целого 0 на выход 0 Бейсик-сценария
  end if

End Sub

```

Рис. 4.114. Использование виртуального тега в редакторе сценариев

4.8.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.115.
2. Выполните *второе действие* из занятия 1. В нашем примере задан период запуска приложения 1 секунда.

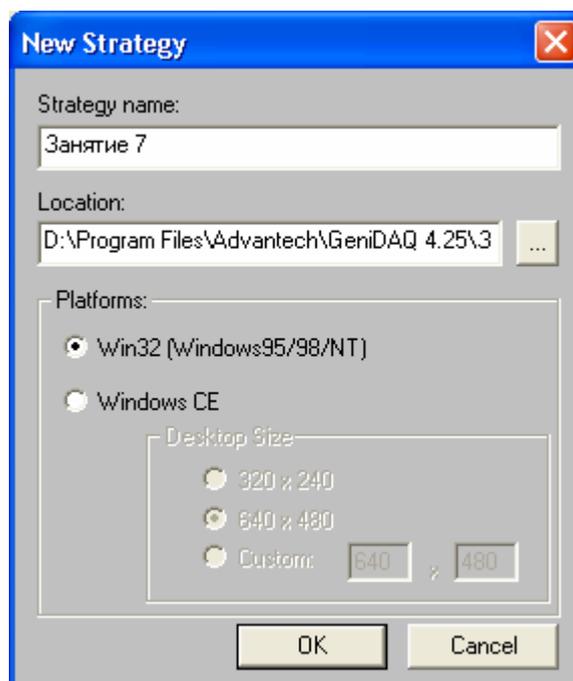


Рис. 4.115. Диалоговое окно конфигурирования приложения

3. *Третье действие.* Сделайте активным окно редактора задач и создайте объект виртуальный тег. Для этого выполните команду **Setup | Add/Delete Virtual Tags** и в появившемся диалоговом окне (см. рис. 4.110) нажмите кнопку **Add Tag**. Появится следующее диалоговое окно и с его помощью сконфигурируйте виртуальный тег в соответствии с рис. 4.111.
4. *Четвертое действие.* В окно редактора задач поместите функциональные блоки аналогового ввода, Бейсик-сценария и тег. Произведите двойной щелчок левой кнопкой мыши, поместив курсор на изображение блока аналогового ввода, и выполните его настройку на ввод информации от канала 0 программного эмулятора **Advantech DEMO I/O** в соответствии с рис. 4.116. Установите связь между функциональным блоком тег и созданным объектом виртуальный тег в соответствии с описанной ранее схемой и рис. 4.112. Произведите двойной щелчок левой кнопкой мыши, поместив курсор на пиктограмму блока Бейсик-сценарий и введите исходный текст программы в соответствии с рис. 4.114. Хотя это и не является обязательным, но, для наглядности, соедините блоки аналогового ввода и тег с функциональным блоком Бейсик-сценария. После этого окно редактора задач приобретет вид, показанный на рис. 4.117.
5. *Пятое действие.* Активизируйте окно редактора форм отображения. Добавьте в это окно элементы отображения индикатор, два цифровых индикатора и две текстовых строки (рис. 4.118). Сконфигурируйте индикатор в соответствии с

рис. 4.119. Сконфигурируйте *цифровые индикаторы* в соответствии с рис. 4.113, рис. 4.120 и текстовые строки в соответствии с рис. 4.118.

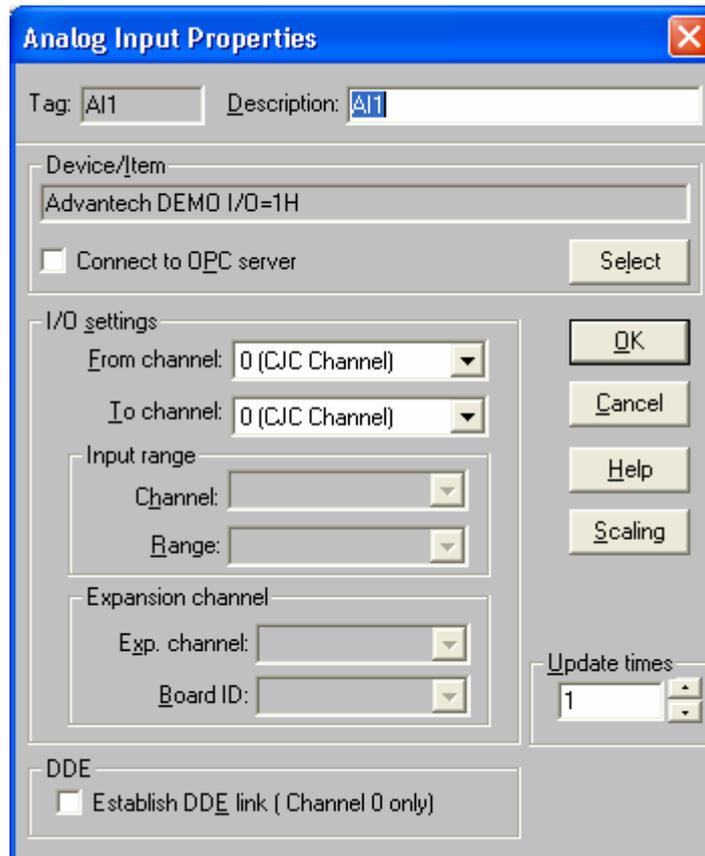


Рис. 4.116. Конфигурирование блока аналогового ввода

6. *Шестое действие.* Сохраните созданный проект (стратегию). Для этого выполните команду **File | Save As**, в появившемся окне диалога укажите необходимую информацию и нажмите кнопку **Сохранить**. Это нужно выполнить для обоих файлов проекта с расширениями **.gni** и **.hld**.
7. *Седьмое действие.* После сохранения файлов созданного проекта запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.121). В процессе работы приложения обратите внимание на состояние индикатора. Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
8. *Восьмое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

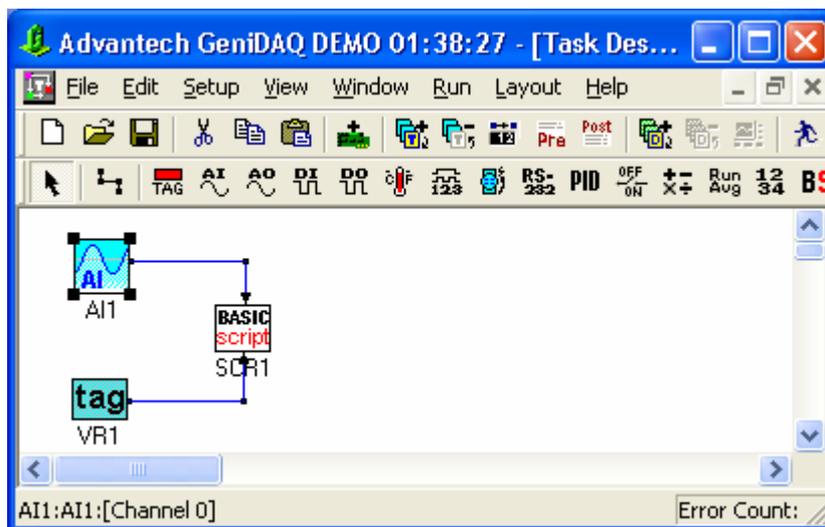


Рис. 4.117. Окно редактора задач

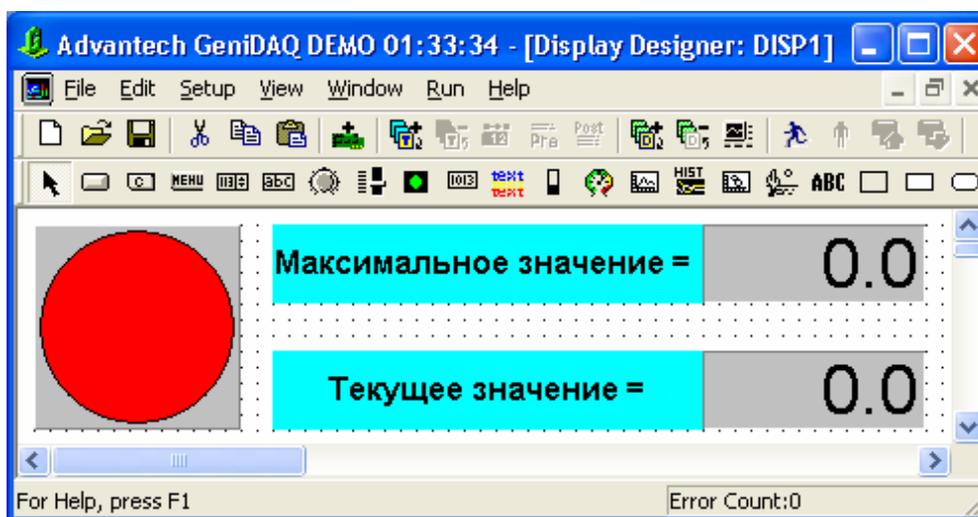


Рис. 4.118. Окно редактора форм отображения

4.8.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

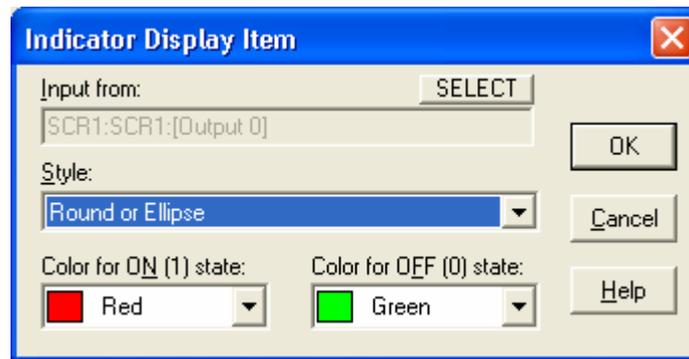


Рис. 4.119. Конфигурирование индикатора

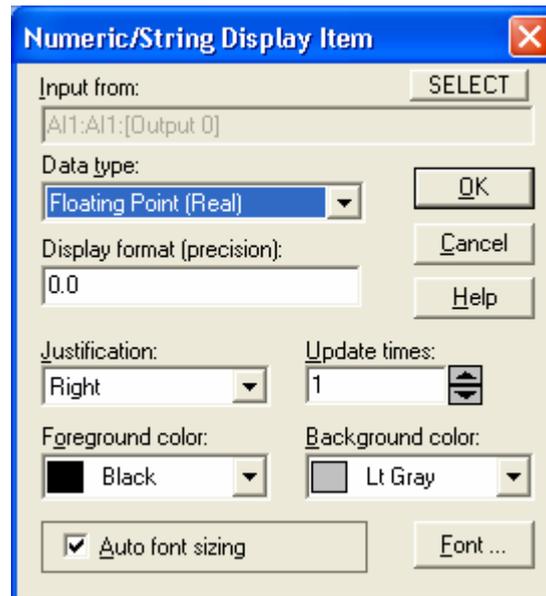


Рис. 4.120. Конфигурирование цифрового индикатора

Упражнение 4.22. Повтор создания и тестирования приложения "Занятие 7"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 7**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

Упражнение 4.23

Разработайте стратегию, обладающую следующими возможностями (в качестве аналога используйте стратегию из занятия 7).

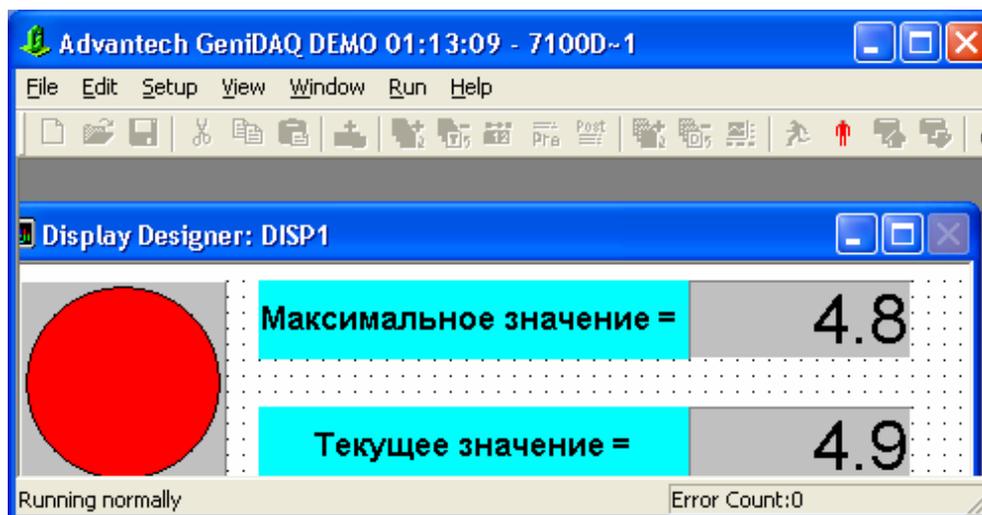


Рис. 4.121. Демонстрация работы проекта в окне отображения

В окно редактора задач поместите функциональный блок аналогового ввода, использующий эмулятор пилообразного сигнала, функциональный блок Бейсик-сценария и два функциональных блока тег, каждый из которых связывается с соответствующим виртуальным тегом. В окне редактора форм отображения поместите три цифровых индикатора. Настройте первый из цифровых индикаторов таким образом, чтобы он отображал сигнал на выходе блока аналогового ввода. Запрограммируйте блок Бейсик-сценария и настройте оставшиеся цифровые индикаторы так, чтобы они работали по следующему алгоритму. Второй цифровой индикатор должен отображать максимальное значение сигнала на выходе блока аналогового ввода, а третий — минимальное значение сигнала на выходе блока аналогового ввода. Все три индикатора в окне отображения снабдить поясняющим текстом.

Упражнение 4.24

Разработайте стратегию, обладающую следующими возможностями (в качестве аналога используйте стратегию из занятия 7). Используйте два окна редактора задач. Придумайте стратегию, в которой происходит передача информации между задачами с использованием виртуальных тегов. Сконфигурируйте окно отображения для подтверждения указанной передачи.

4.9. Занятие 8 "Программирование основного сценария"

Цель занятия состоит в изучении базовых правил разработки *основного сценария* и способов его применения для управления задачами. С этой целью в рамках стратегии

создается задача и основной сценарий для управления этой задачей. Используются блок аналогового ввода (**Analog Input**) редактора задач, инструменты график времени (**Realtime Trend Graph**), цифровой индикатор (**Numeric String**) и текстовые строки (**Text String**) редактора форм отображения, а также *основной сценарий* стратегии.

4.9.1. Используемый инструментарий

Новым инструментом на данном занятии является *основной сценарий*. Особенности его использования описаны в разд. 4.6 "Использование языка VBA в SCADA-системах".

4.9.2. Проектирование приложения

Проектирование приложения. Для реализации поставленного задания выполните следующие действия.

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.122.
2. *Второе действие.* Сделайте активным окно редактора задач и поместите в него функциональный блок аналогового ввода. Произведите двойной щелчок левой кнопкой мыши, поместив курсор на изображение блока *аналогового ввода* в окне редактора задач, и выполните его настройку на ввод информации от канала 0 программного эмулятора **Advantech DEMO I/O** в соответствии с приведенным ранее рис. 4.116.
3. *Третье действие.* Активизируйте окно редактора форм отображения. Добавьте в это окно элементы отображения график времени, цифровой индикатор и текстовую строку (рис. 4.123). Сконфигурируйте *график времени* и *цифровой индикатор* в соответствии с рис. 4.124 и 4.125.
4. *Четвертое действие.* Задайте временные параметры запуска задачи **TASK1** таким образом, чтобы ее запуск осуществлялся под управлением основного сценария. Для этого выполните команду **Setup | Task Properties...** и настройте параметры запуска задачи в соответствии с рис. 4.126. При этом будет задан пассивный запуск задачи по команде из основного сценария.
5. *Пятое действие.* Добавьте в стратегию основной сценарий. Для этого выполните команду **File | Add/Delete | Add Main Script** и в появившемся окне редактора сценариев введите текст, показанный на рис. 4.127.

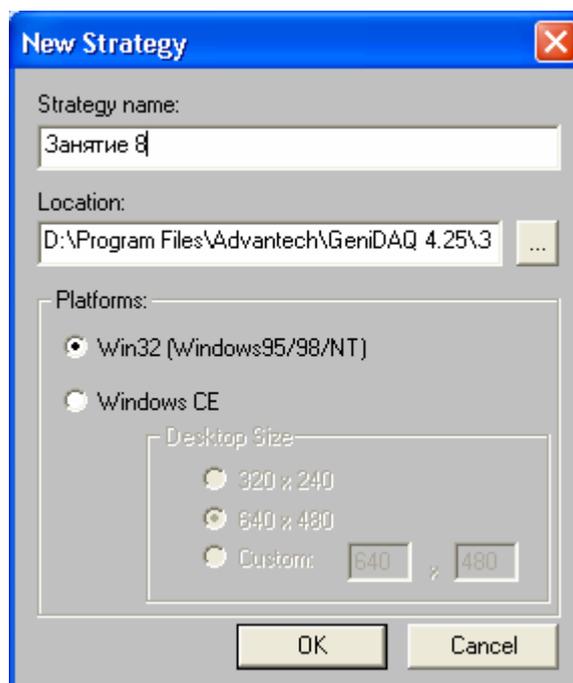


Рис. 4.122. Диалоговое окно конфигурирования приложения

6. *Пятое действие.* Сохраните созданный проект (стратегию). Для этого выполните команду **File | Save As**, в появившемся окне диалога укажите необходимую информацию и нажмите кнопку **Сохранить**. Это нужно выполнить для обоих файлов проекта с расширениями *.gni* и *.hld*.
7. *Седьмое действие.* После сохранения файлов созданного проекта запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.128). Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
8. *Восьмое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

4.9.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

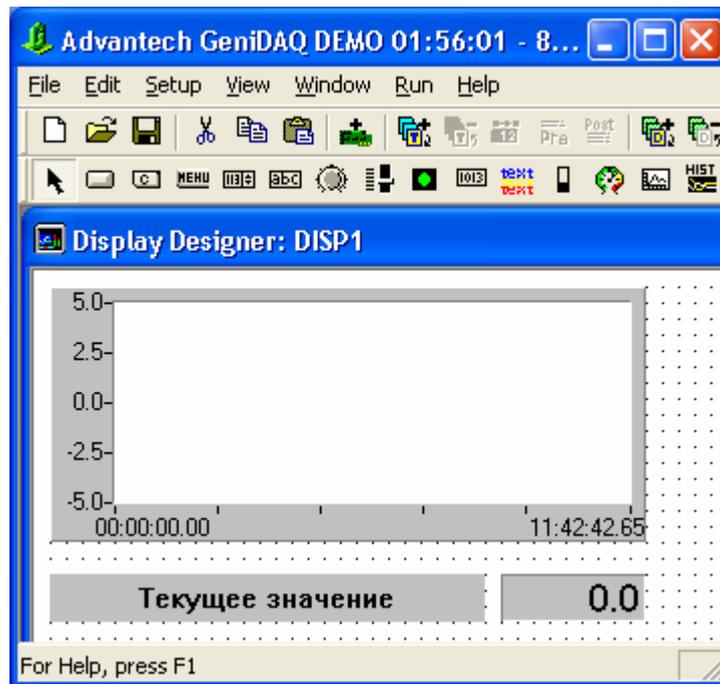


Рис. 4.123. Окно отображения

Упражнение 4.25. Повтор создания и тестирования приложения "Занятие 8"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 8**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

4.10. Занятие 9 "Управление несколькими задачами"

Цель занятия состоит в изучении способа *управления несколькими задачами* в рамках одной стратегии с помощью настройки параметров выполнения каждой задачи. С этой целью создаются две отдельные задачи, каждая из которых содержит блок аналогового ввода. Создается окно отображения, содержащее два элемента отображения временной график, на которые выводятся сигналы с выходов блоков аналогового ввода. Параметры каждой задачи настраиваются таким образом, чтобы они имели разные режимы выполнения.

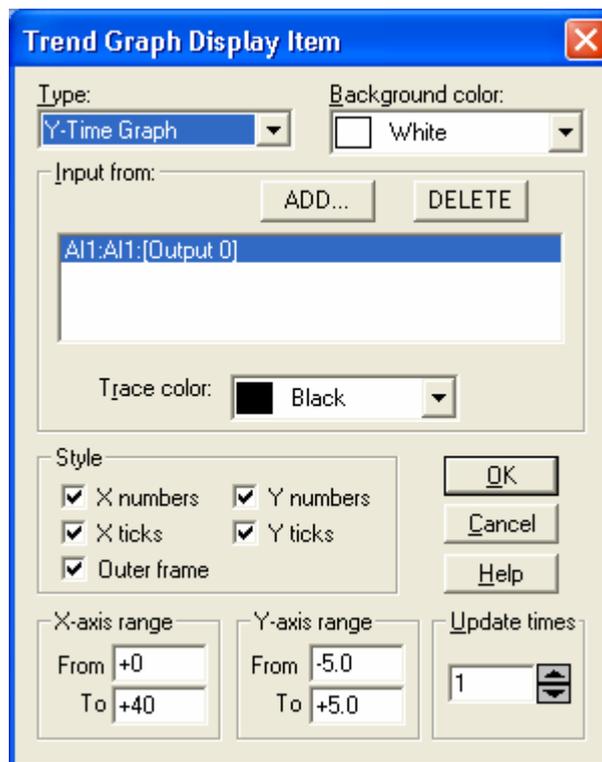


Рис. 4.124. Конфигурирование графика времени

4.10.1. Используемый инструментарий

Новым инструментом на данном занятии является использование диалогового окна **Scan Task Setup**, с помощью которого задаются параметры запуска задачи (рис. 4.129) — интервал запуска, способ запуска и продолжительность выполнения.

4.10.2. Проектирование приложения

1. Выполните *первое действие* из занятия 1. Создаваемое приложение настройте в соответствии с рис. 4.130.
2. *Второе действие*. Сделайте активным окно **TASK1** редактора задач и поместите в него функциональный блок аналогового ввода. Произведите двойной щелчок левой кнопкой мыши, поместив курсор на изображение блока аналогового ввода в окне редактора задач, и выполните его настройку на ввод информации от канала 0 программного эмулятора **Advantech DEMO I/O** в соответствии с приведенным ранее рис. 4.116. Добавьте в стратегию вторую задачу **TASK2**, для чего выполните команду **File | Add/Delete | Add Task**. Аналогичным образом поместите в него функциональный блок аналогового ввода. Произведите двойной

щелчок левой кнопкой мыши, поместив курсор на изображение блока аналогового ввода в окне редактора задач, и выполните его настройку на ввод информации от канала 1 программного эмулятора **Advantech DEMO I/O** в соответствии с рис. 4.131.

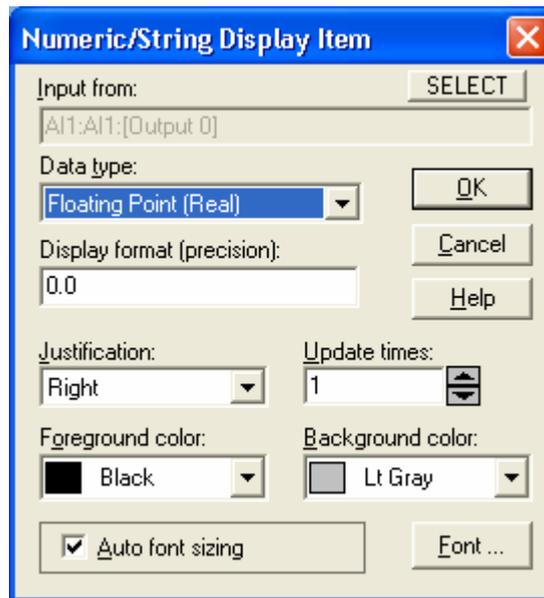


Рис. 4.125. Конфигурирование цифрового индикатора

3. *Третье действие.* Активизируйте окно редактора форм отображения. Добавьте в это окно два элемента отображения график времени и две текстовые строки (рис. 4.132). Сконфигурируйте графики времени в соответствии с рис. 4.133 и 4.134. При этом на одном из них будет отображаться информация с выхода блока аналогового ввода задачи **TASK1**, а на втором — задачи **TASK2**.
4. *Четвертое действие.* Задайте временные параметры запуска задачи **TASK1**. Сделайте активным окно задачи **TASK1**, для чего поместите курсор во внутреннюю область ее окна и произведите щелчок левой кнопкой мыши. Выполните команду **Setup | Task Properties... (Ctrl+T)** задайте параметры задачи в соответствии с рис. 4.135. Такая настройка обеспечивает запуск задачи в 16 часов 28 минут и ее выполнение в течение 15 секунд. Аналогичным образом задайте временные параметры запуска задачи **TASK2**. Сделайте активным окно задачи **TASK2**, для чего поместите курсор во внутреннюю область ее окна и произведите щелчок левой кнопкой мыши. Выполните команду **Setup | Task Properties... (Ctrl+T)**, задайте параметры задачи в соответствии с рис. 4.136. Такая настройка обеспечивает запуск задачи **TASK2** по истечении 15 секунд от момента запуска всей стратегии с ее остановкой после 10 циклов выполнения.

5. *Пятое действие.* Сохраните созданный проект (стратегию) аналогично тому, как это делалось на занятии 1. Для этого выполните команду **File | Save As**, в появившемся окне диалога укажите необходимую информацию и нажмите кнопку **Сохранить**. Это нужно выполнить для обоих файлов проекта с расширениями *.gni* и *.HLD*.

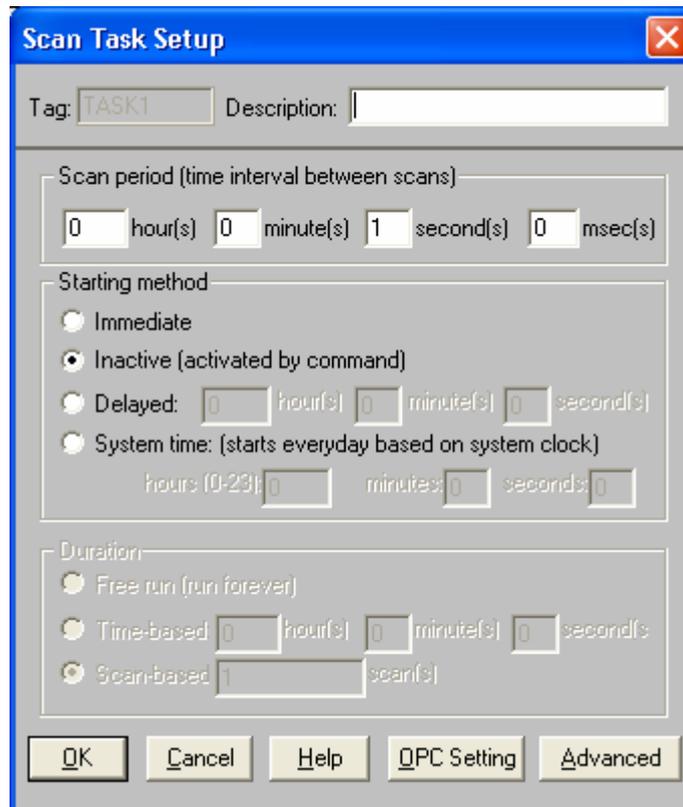


Рис. 4.126. Задание временных параметров запуска задачи

6. *Шестое действие.* После сохранения файлов созданного проекта запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.137). Обратите внимание на особенности запуска и останова задач. Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов.
7. *Седьмое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

4.10.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

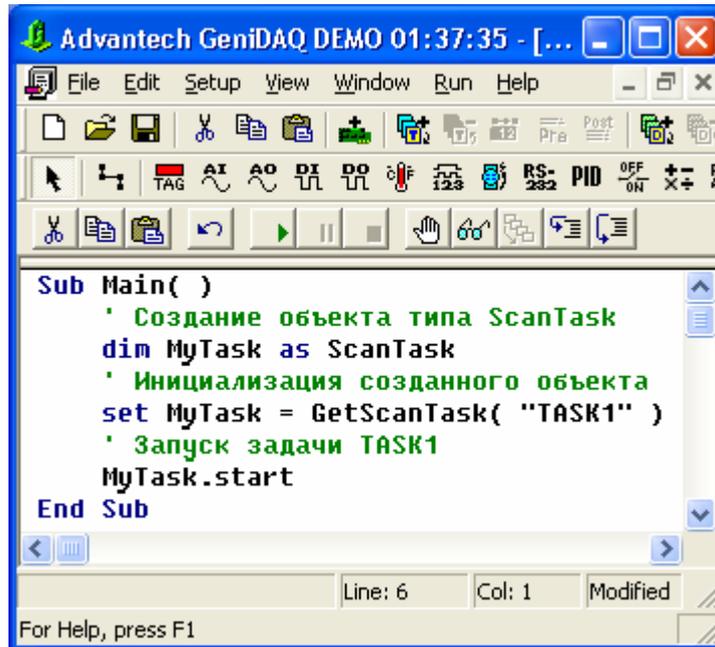


Рис. 4.127. Основной сценарий стратегии

Упражнение 4.26. Повтор создания и тестирования приложения "Занятие 9"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 9**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

Упражнение 4.27

Разработайте стратегию, обладающую следующими возможностями. Создайте две задачи. В окно редактора задач каждой из задач поместите функциональный блок аналогового ввода, использующий эмулятор гармонического сигнала и прямоугольного сигнала. Настройте параметры запуска задач таким образом, чтобы период их выполнения составлял 1 секунду, обеспечьте запуск третьей задачи с задержкой 10 секунд и ее выполнение 15 раз и пассивный запуск второй задачи с помощью главного сценария. В окно редактора форм отображения поместите два

графика и настройте их таким образом, чтобы они отображали сигналы на выходах соответствующих блоков аналогового ввода.

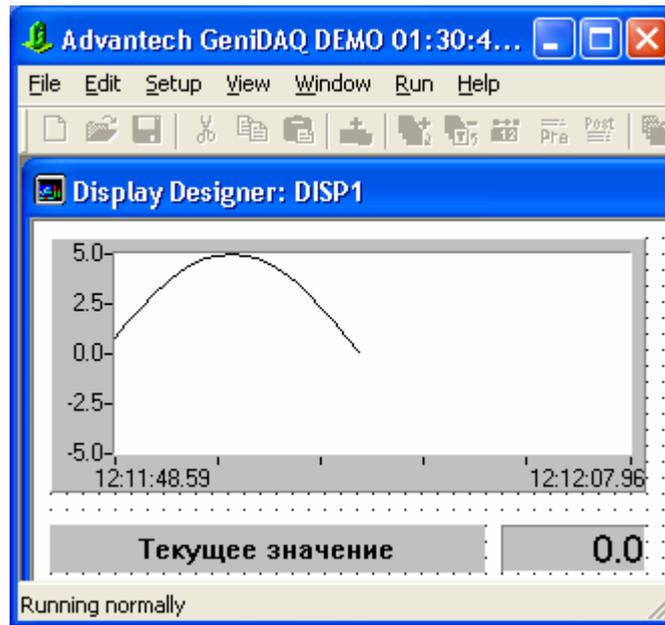


Рис. 4.128. Демонстрация работы проекта в окне отображения

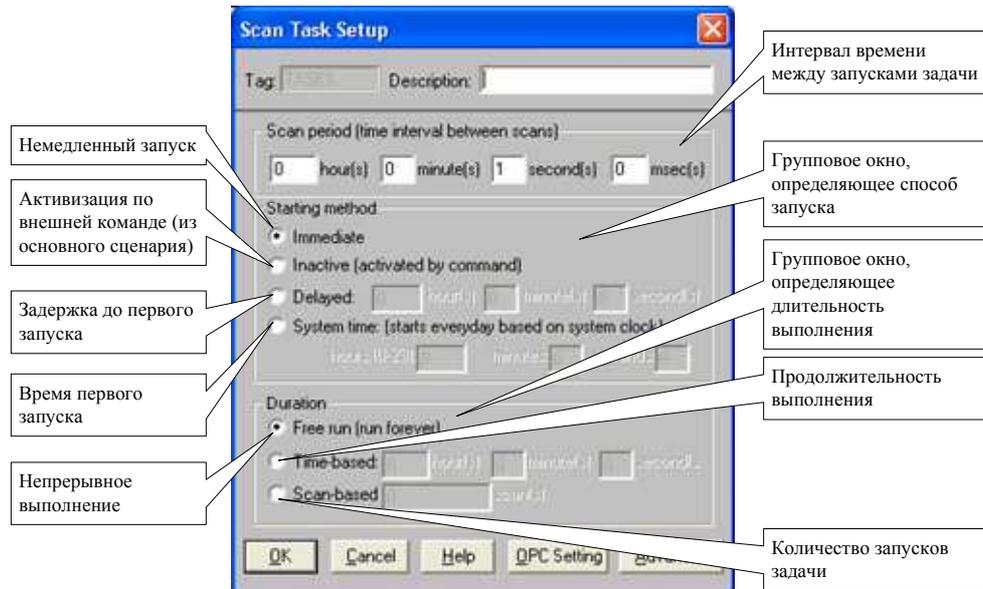


Рис. 4.129. Диалоговое окно задания параметров запуска задачи

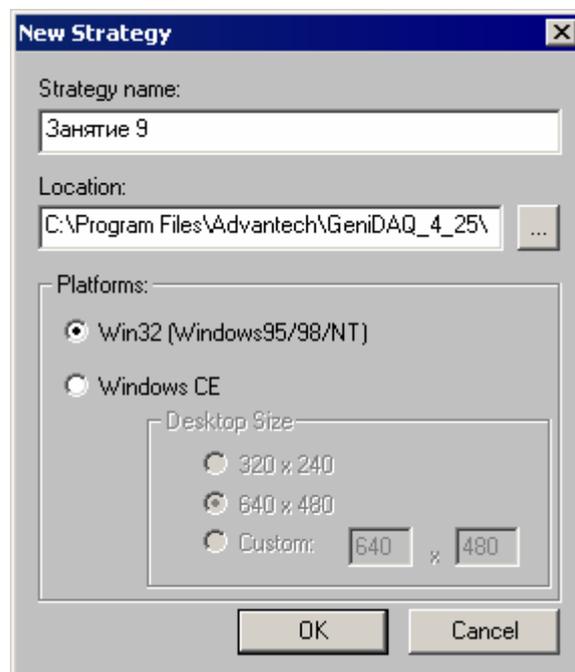


Рис. 4.130. Диалоговое окно конфигурирования приложения

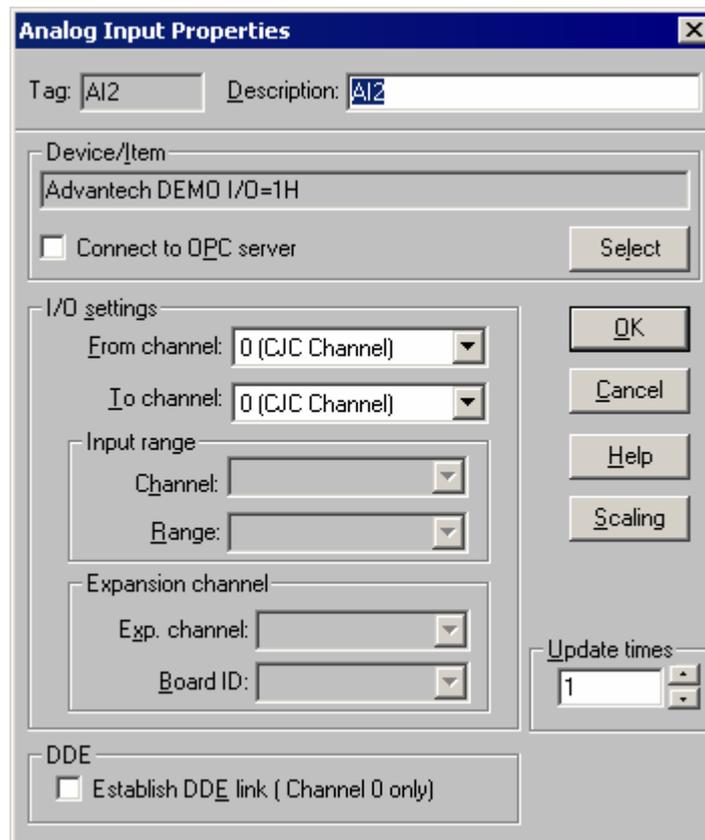


Рис. 4.131. Конфигурирование блока аналогового ввода задачи TASK2

Упражнение 4.28

Разработайте стратегию, обладающую следующими возможностями. Создайте две задачи. В окне редактора задач каждой из задач поместите функциональный блок аналогового ввода, использующий эмулятор гармонического сигнала и прямоугольного сигнала. Настройте параметры запуска задач таким образом, чтобы период их выполнения составлял 1 секунду, обеспечьте запуск третьей задачи с задержкой 10 секунд и ее выполнение 15 раз и запуск второй задачи в заданное время и ее выполнение в течение 30 секунд. В окне редактора форм отображения поместите два графика и настройте их таким образом, чтобы они отображали сигналы на выходах соответствующих блоков аналогового ввода.

4.11. Занятие 10 "DDE-обмен с использованием блока Бейсик-сценария редактора задач"

Цель занятия состоит в изучении способа динамического обмена данными между созданной стратегией (источник) и внешним приложением (электронная таблица, приемник). Динамический обмен данными (DDE-обмен) реализуется с помощью функционального блока Бейсик-сценария редактора задач, предварительного и пост сценариев задачи. С этой целью в рамках стратегии создается задача, содержащая блок аналогового ввода, виртуальный тег и блок Бейсик-сценария. В папку сценария помещается файл электронной таблицы, в который созданное приложение в процессе работы выводит результаты измерений, выполненных блоком аналогового ввода. Описания функционального блока Бейсик-сценария редактора задач, предварительного и пост сценария задачи приведены в занятии 6 и разд. 4.6. Виртуальный тег описан в занятии 7.

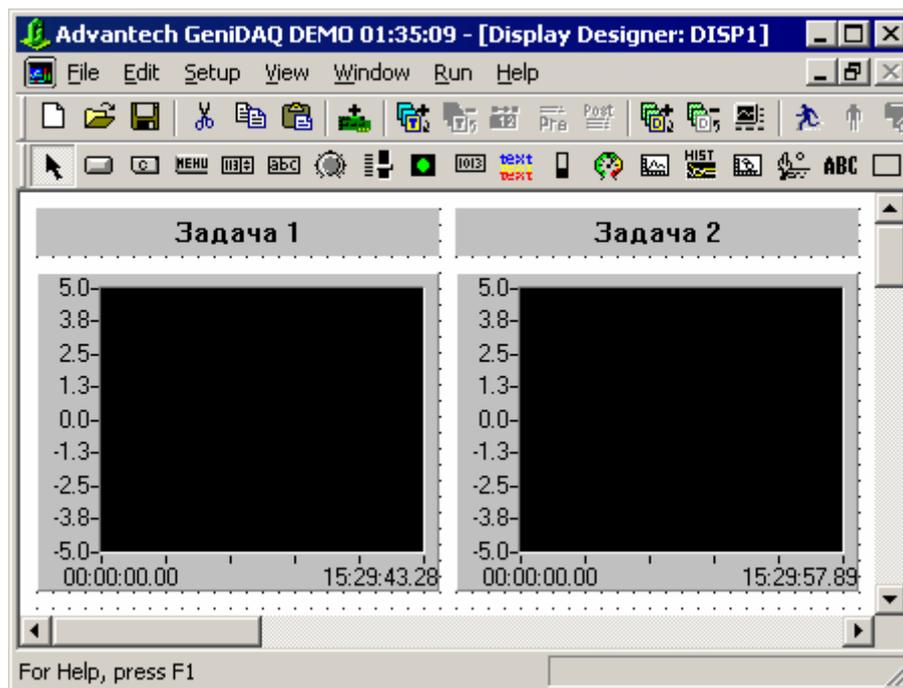


Рис. 4.132. Конфигурация окна отображения

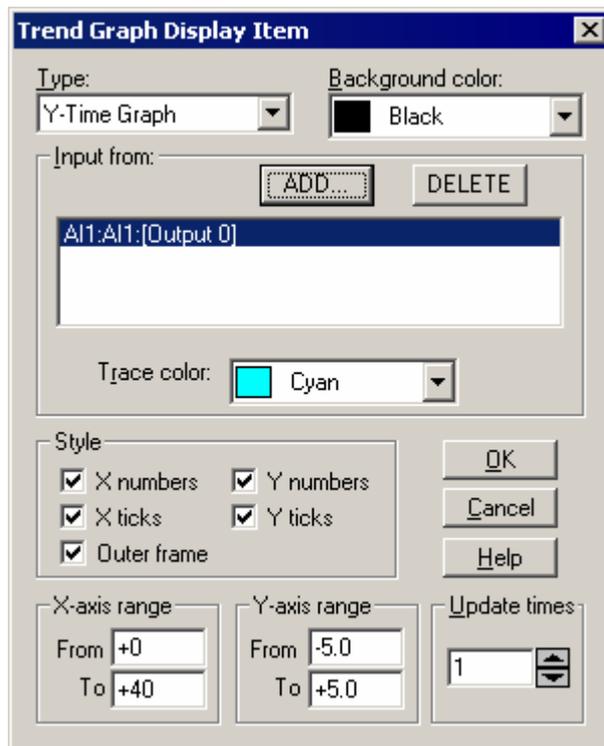


Рис. 4.133. Конфигурация графика для блока аналогового ввода задачи **TASK1**

4.11.1. Проектирование приложения

1. Выполните *первое действие* из занятия 1. Создаваемое приложение назовите **Занятие 10**.
2. *Второе действие*. Сделайте активным окно **TASK1** редактора задач и поместите в него функциональные блоки аналогового ввода, тега и Бейсик-сценария. Выполните настройку блока аналогового ввода в соответствии с приведенными ранее рис. 4.116, добавьте виртуальный тег (команда **Setup | Add/Delete Virtual Tags | Add Tag**) и настройте его в соответствии с рис. 4.138, настройте тег и блок Бейсик-сценария в соответствии с рис. 4.139 и 4.140. Полный текст функционального блока Бейсик-сценария, снабженный подробными комментариями, вполне ясен:

```
' Вывод из стратегии (приложения) в электронную таблицу с
' использованием DDE-технологии
sub SCR1( )
' Определение объектов
Dim index As Tag
Dim mytag As Tag
Dim i As Integer
```

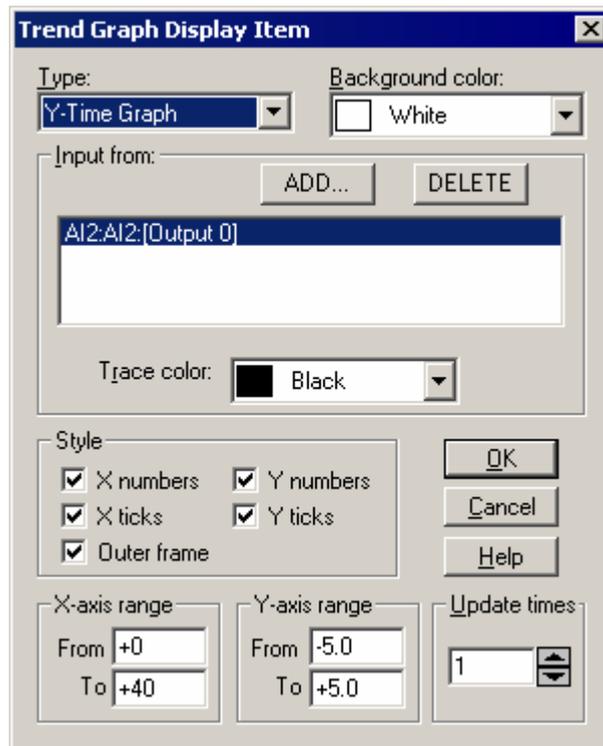


Рис. 4.134. Конфигурация графика для блока аналогового ввода задачи TASK2

```

' Инициализация DDE связи с электронной таблицей (Excel)
NumCan% = DDEInitiate( "Excel", "[Book1.xls]Лист1" )
' Выше:
'   использование NumCan% (уникальный номер канала)
'   эквивалентно Dim NumCan as Integer;
'   "Excel" задает приложение, с которым устанавливается
'   связь;
'   "[Book1.xls]Лист1" задает файл (Book1.xls), с
'   которым будет работать электронная таблица и его
'   текущую вкладку (Лист1)
' Получение значения объекта для виртуального тега VT1
Set index = GetTag( "VIRTASK", "VT1" )
' Получение значения объекта для функционального блока
' аналогового ввода AI1
Set mytag = GetTag( "TASK1", "AI1" )
' Сохранение данных из AI1 в электронной таблице
DDEPoke NumCan%, "R1"+"C"+Format$( index.value ), _
    Format$( mytag.value )
' Выше:
'   Format$( index.value ), Format$( mytag.value )
'   преобразуют значение своего аргумента в строку;

```

Scan Task Setup

Tag: Description:

Scan period (time interval between scans)

hour(s) minute(s) second(s) msec(s)

Starting method

Immediate

Inactive (activated by command)

Delayed: hour(s) minute(s) second(s)

System time: (starts everyday based on system clock)

hours (0-23): minutes: seconds:

Duration

Free run (run forever)

Time-based: hour(s) minute(s) second(s)

Scan-based scan(s)

Рис. 4.135. Конфигурирование параметров выполнения задачи **TASK1**

```

' "R1" задает вывод в строку 1 файла электронной
' таблицы (Row 1);
' "C"+Format$( index.value ) задает номер столбца (Col)
' электронной таблицы, + операция конкатенации строк;
' Format$( mytag.value ) задает значение, выводимое в
' ячейку электронной таблицы
' Разрыв DDE связи
DDETerminate NumCan%
' Изменение значения index, определяющего номер столбца в
' файле электронной таблицы
i = index.value
If index.value >= 10 Then
    index.value = 1
Else
    index.value = i + 1
End If
End Sub

```

Scan Task Setup

Tag: Description:

Scan period (time interval between scans)

hour(s) minute(s) second(s) msec(s)

Starting method

Immediate

Inactive (activated by command)

Delayed: hour(s) minute(s) second(s)

System time: (starts everyday based on system clock)

hours (0-23): minutes: seconds:

Duration

Free run (run forever)

Time-based: hour(s) minute(s) second(s)

Scan-based scan(s)

Рис. 4.136. Конфигурирование параметров выполнения задачи TASK2

3. *Третье действие.* Задайте временные параметры запуска задачи **TASK1**. Выполните команду **Setup |Task Properties...** (**Ctrl+T**) задайте параметры задачи, соответствующие умолчанию (период повторения 1 секунда).
4. *Четвертое действие.* Задайте предварительный и пост сценарии задачи в соответствии с рис. 4.122 и 4.123. Полный текст предварительного сценария задачи, снабженный подробными комментариями, также вполне ясен и приведен далее.

' Пост-сценарий задачи

Sub POST_TASK1()

' Определение объекта

Dim excel **As** Object

' Получение значения объекта - электронной таблицы

Set excel = GetObject(, "Excel.Application")

' Закрываем файл book1.xls с его сохранением и

' завершение работы электронной таблицы

excel.Workbooks("book1.xls").Close SaveChanges := True

excel.Quit

End Sub

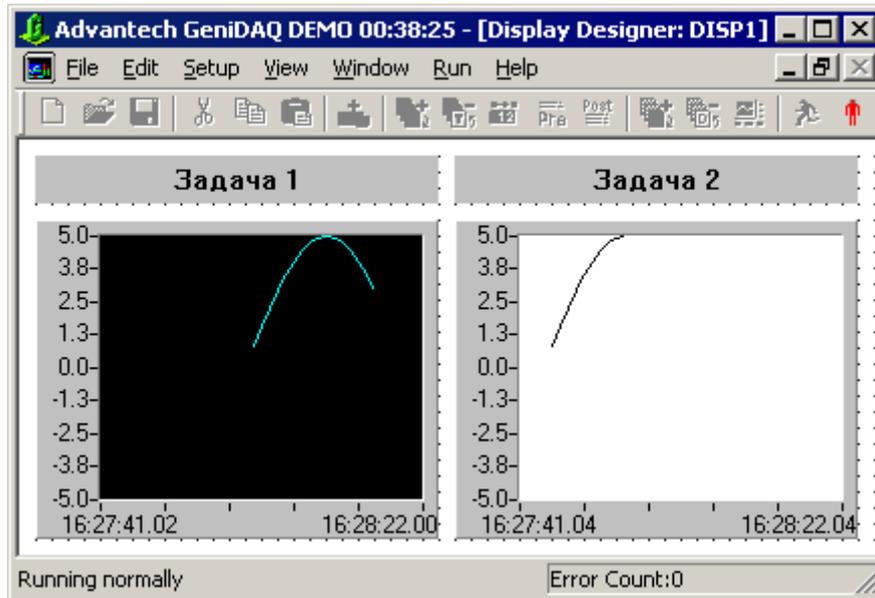


Рис. 4.137. Демонстрация работы проекта в окне отображения

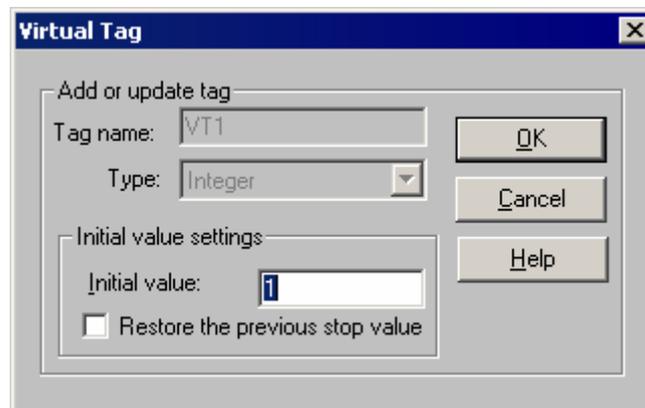


Рис. 4.138. Конфигурирование виртуального тега

5. *Пятое действие.* С помощью приложения **Microsoft Excel** создайте файл Book1.xls в папке проекта.
6. *Шестое действие.* Сохраните созданный проект (стратегию) аналогично тому, как это делалось на занятии 1.
7. *Седьмое действие.* После сохранения файлов созданного проекта запустите созданный проект с помощью кнопки **Start** на панели инструментов (рис. 4.143). Обратите внимание на то, что первом плане отображается электронная таблица в

процессе ее заполнения. Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов, предварительно перейдя в окно SCADA системы.

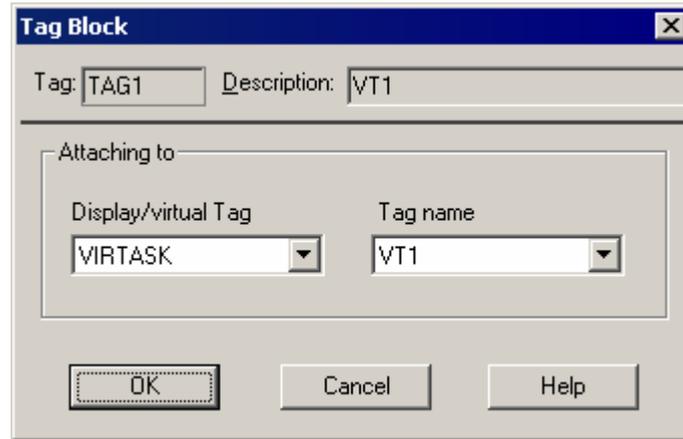


Рис. 4.139. Конфигурирование тега

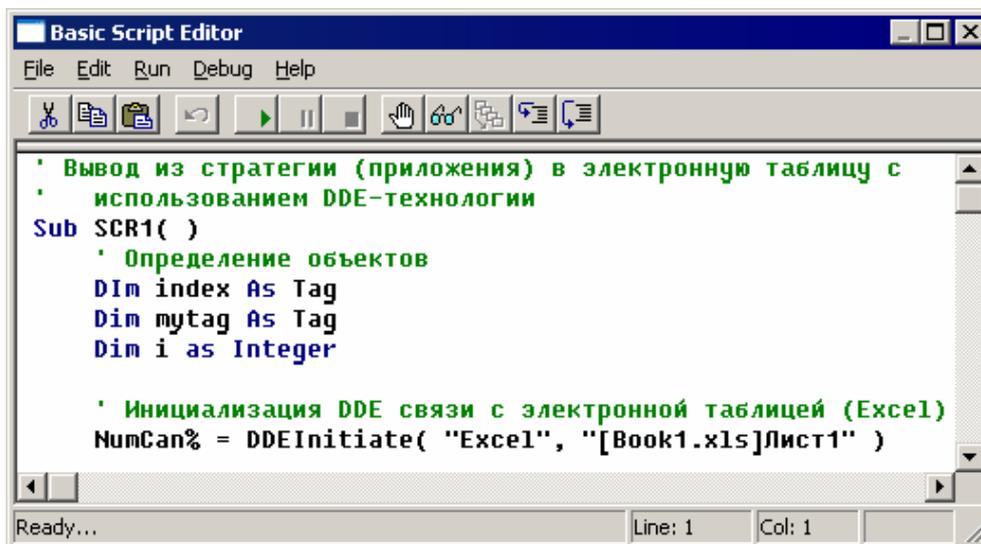


Рис. 4.140. Функциональный блок Бейсик-сценария

8. *Восьмое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

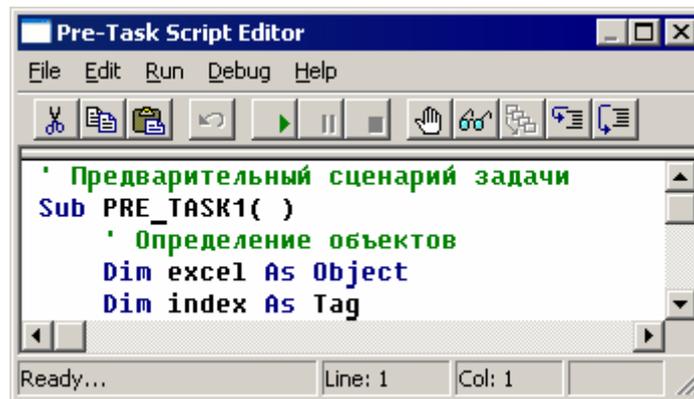


Рис. 4.141. Предварительный сценарий задачи TASK1

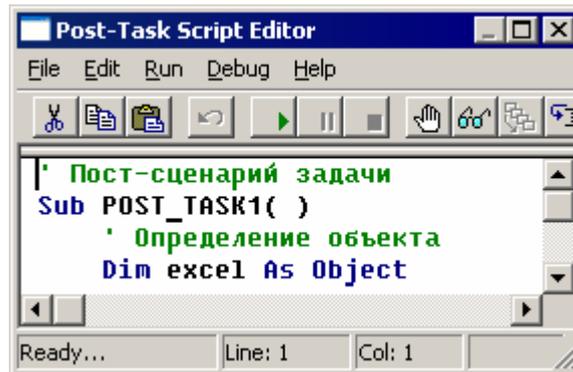


Рис. 4.142. Пост сценарий задачи TASK1

The screenshot shows an Excel spreadsheet window titled 'Book1'. The spreadsheet has columns labeled A through J and rows labeled 1 through 4. The data in row 1 is as follows:

	A	B	C	D	E	F	G	H	I	J
1	-0,78218	-1,54509	-2,26996	-2,93893	3,535535	2,938925	2,269955	1,545085	0,78217	0
2										
3										
4										

The status bar at the bottom shows 'Лист1 / Лист2 / Лист3 /'.

Рис. 4.143. Горизонтальное заполнение электронной таблицы

4.11.2. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.29. Повтор создания и тестирования приложения "Занятие 10"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 10**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

Упражнение 4.30

Разработайте стратегию, обладающую следующими возможностями. Стратегия (приложение) должна выводить в электронную таблицу (файл "Пример.xls") выходной сигнал блока аналогового ввода, использующий эмулятор пилообразного сигнала. Вывод в электронную таблицу выполнить циклически по главной диагонали (15 значений по кольцу).

Упражнение 4.31

Разработайте стратегию, обладающую следующими возможностями. Стратегия (приложение) должна выводить в электронную таблицу (файл "Прямоугольник.xls") выходной сигнал блока аналогового ввода, использующий эмулятор прямоугольного сигнала. Вывод в электронную таблицу выполнить циклически по вспомогательной диагонали (12 значений по кольцу).

Упражнение 4.32

Разработайте стратегию, обладающую следующими возможностями. Стратегия (приложение) должна выводить в электронную таблицу (файл "Пила.xls") выходной сигнал блока аналогового ввода, использующий эмулятор пилообразного сигнала. Вывод в электронную таблицу выполнить циклически по сторонам квадрата размером 8 по часовой стрелке.

Упражнение 4.33

Разработайте стратегию, обладающую следующими возможностями. Стратегия (приложение) должна выводить в электронную таблицу (файл "Пила.xls") выходной сигнал блока аналогового ввода, использующий эмулятор гармонического сигнала. Вывод в электронную таблицу выполнить циклически по сторонам квадрата размером 10 против часовой стрелки.

4.12. Занятие 11 "Работа с таймером"

Цель занятия состоит в изучении правил работы с *таймером*. С этой целью создается стратегия, использующая *функциональные блоки таймера (Timer)* и *метки времени (Time Stamp)* редактора задач.

4.12.1. Используемый инструментарий

Новыми инструментами на данном занятии являются функциональные блоки таймера и метки времени.

Функциональный блок таймера редактора задач предназначен для реализации таймеров различных типов и имеет вход сброса. Таймер может выполнять функции формирования абсолютных или относительных временных интервалов с разрешением 0.1 секунды или 1 секунда. Цикл таймера до сброса с последующим возобновлением работы может составлять от 1 минуты до 1 года. Содержимое таймера (относительное или абсолютное время) может быть передано другому функциональному блоку стратегии. Выходное значение блока таймера представляется в виде длинного целого значения в диапазоне от 0 до 4294967295. Данный блок является весьма удобным средством для реализации различных алгоритмов управления, выполнение которых основывается на интервалах времени. Диалоговое окно настройки параметров блока показано на рис. 4.144. Единицы, в которых выражается значение на выходе блока таймера, могут быть выбраны с помощью переключателя *разрешающая способность (Resolution)*. Таким образом, разрешающая способность таймера может составлять 1 тик (0.1 секунды) или 1 секунда. Получение более высокого разрешения в рамках операционной системы Windows представляется весьма трудной задачей. Цикличность (периодичность) работы таймера может быть выбрана с помощью переключателя *цикл (Cycle)* диалогового окна настройки параметров блока. По истечении периода времени, выбранного с помощью указанного переключателя, содержимое таймера сбрасывается в ноль и работа таймера возобновляется. Например, если выбран цикл, равный 1 минуте (**By minute**), а разрешение (**Resolution**) — 1 секунда, то содержимое таймера в каждом цикле будет увеличиваться от 0 до 59 с последующим возобновлением. Данная функция блока таймера в текущей версии пакета работает только при использовании разрешающей способности, равной 1 секунде. Содержимое блока таймера может быть сброшено в 0 в процессе его работы путем подачи на вход блока дискретного сигнала с уровнем логической единицы. Для возобновления работы таймера следует подать на вход блока дискретный сигнал с уровнем логического нуля.

При использовании таймера, отсчитывающего абсолютное время (переключатель *тип таймера — Timer type* — установлен в положение *прошедшее время — Elapsed time*), после запуска стратегии на исполнение алгоритм работы таймера объясняется следующим примером. Если разрешающая способность таймера составляет 1 секунду, а в качестве цикла таймера выбрана 1 минута, то при запуске стратегии на исполнение в 11:23:17 (по часам реального времени компьютера) работа таймера начнется со значения, равного 17, и будет продолжаться 59 секунд последующим сбросом и дальнейшим возобновлением работы. Нулевые начальные значения содержимого таймера при использовании указанного режима работы приведены в табл. 4.19.

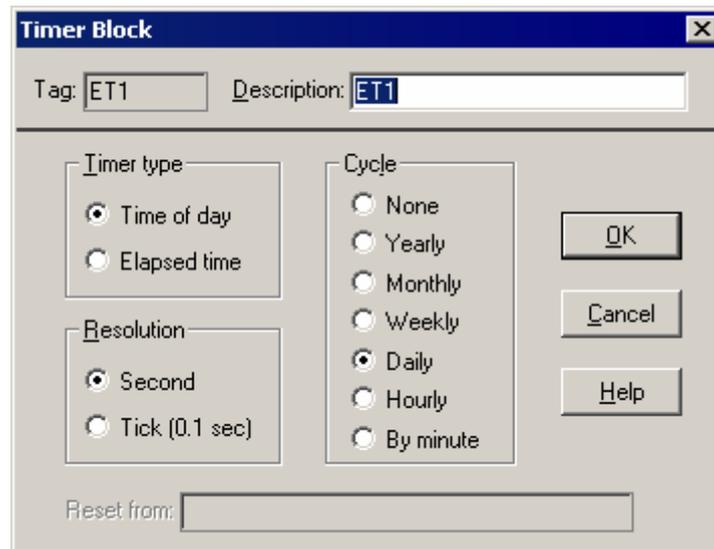


Рис. 4.144. Диалоговое окно для конфигурирования функционального блока таймера

Таблица 4.19. Нулевые начальные значения функционального блока таймера

Цикл таймера	Значение системного времени для нулевого начального содержимого таймера
Год (yearly)	00:00:00, 1 января каждого года
Месяц (monthly)	00:00:00, в первый день каждого месяца
Неделя (weekly)	00:00:00, в воскресенье каждой недели
Сутки (daily)	00:00:00, в начале каждых суток
Час (hourly)	Каждый час
Минута (by minute)	Каждую минуту

Таким образом, имеется возможность вычисления текущего времени. Например, таймер с циклом *неделя* и разрешением 1 секунда при запуске стратегии в 0:00 в понедельник будет иметь начальное содержимое, равное 86400, а в полдень (12:00) пятницы его содержимое увеличится до 518400. В результате появляется возможность автоматического отключения какой-либо единицы контролируемого оборудования в течение выходных. Блок таймера имеет один дискретный вход, предназначенный для сброса содержимого таймера с последующим возобновлением

его работы, и один выход, по которому выводится абсолютное или относительное значение времени (длинного целого типа) в заданных единицах разрешения таймера.

Функциональный блок *метка времени* имеет единственный выход, присоединение которого к элементу отображения или блоку архивации данных позволяет получать значение системного времени в виде строки символов. Имеется возможность использования различных форматов представления системного времени (рис. 4.145).

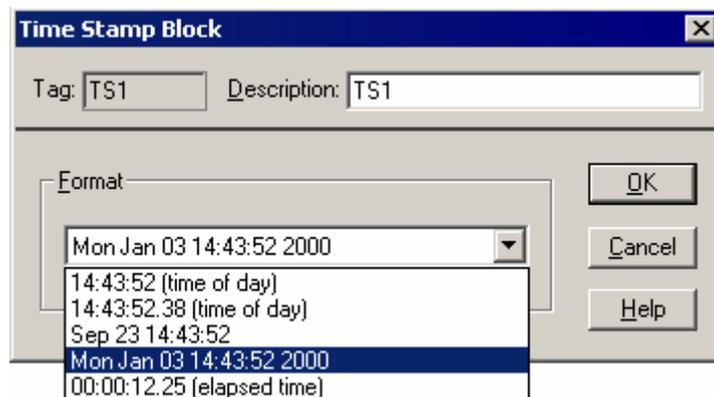


Рис. 4.145. Диалоговое окно для конфигурирования метки времени

Входная связь блока отсутствует — при попытке присоединения к блоку метки времени проводника от другого функционального блока на экран монитора будет выведено сообщение **Вход недоступен (Cannot accept input)**. Выходная связь блока позволяет передавать другим функциональным блокам стратегии значение системного времени в виде строки символов или в ином формате (рис. 4.146).

4.12.2. Проектирование приложения

1. Выполните *первое действие* из занятия 1. Создаваемое приложение назовите **Занятие 11**.
2. *Второе действие*. Сделайте активным окно редактора задач и поместите в него функциональные блоки таймера, реализации простой операции деления, Бейсик-сценария и метки времени, соединив их в соответствии с рис. 4.129. Настройте функциональные блоки в соответствии с рис. 4.144, 4.148 — 4.153 и 4.145. Такая настройка блоков обеспечивает на выходе блока деления **SOC1** значение текущего часа, на выходе **SOC2** — текущей минуты, на выходе **ET3** — текущей секунды, а на выходе **ET4** — текущего тика текущей секунды.
3. *Третье действие*. Активизируйте окно редактора форм отображения. Добавьте в это окно элементы отображения в соответствии с рис. 4.154. Сконфигурируйте их в соответствии с рис. 4.155 — 4.159.

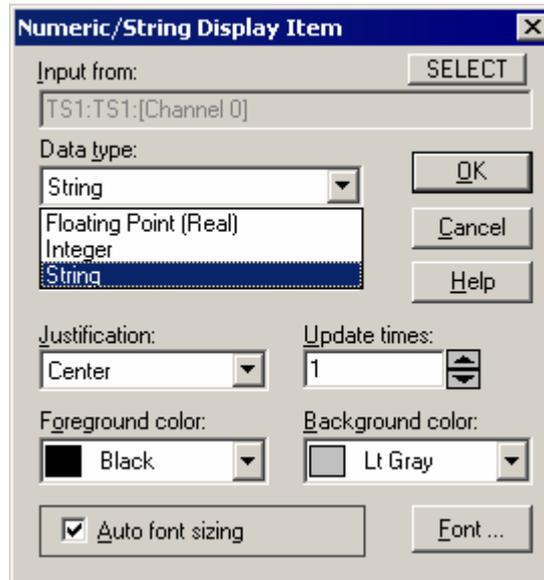


Рис. 4.146. Форматы выдачи информации из блока метки времени в другие блоки

4. *Четвертое действие.* Задайте временные параметры запуска задачи — непрерывный режим работы с периодом перезапуска 100 миллисекунд.
5. *Пятое действие.* Сохраните созданный проект (стратегию) аналогично тому, как это делалось на занятии 1.
6. *Шестое действие.* После сохранения файлов созданного проекта запустите созданный проект с помощью кнопки <Start> на панели инструментов (рис. 4.160). Остановить работу проекта можно с помощью кнопки <Stop> на панели инструментов.
7. *Седьмое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.

4.11.2. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.34. Повтор создания и тестирования приложения "Занятие 11"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 11**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

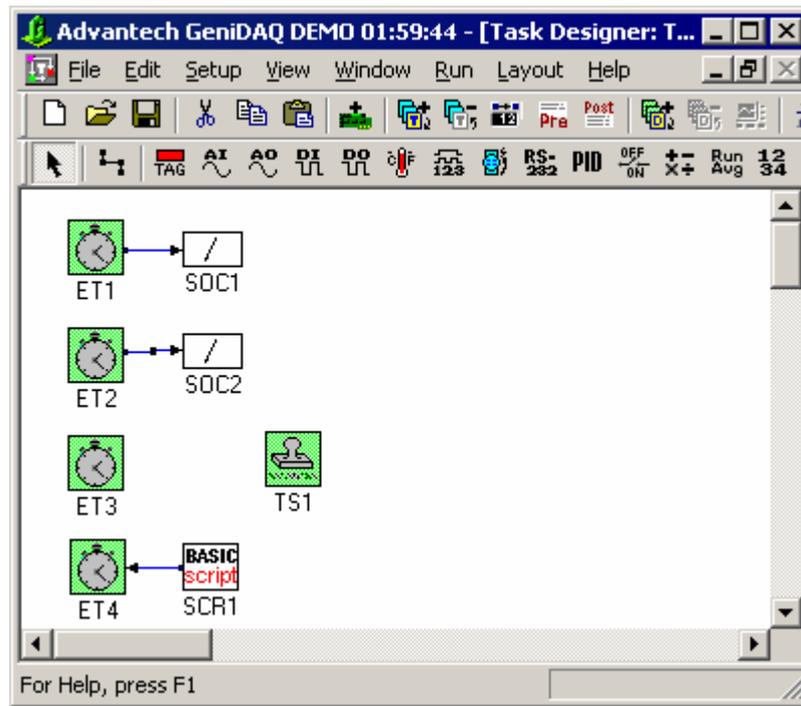


Рис. 4.147. Окно редактора задач с функциональными блоками

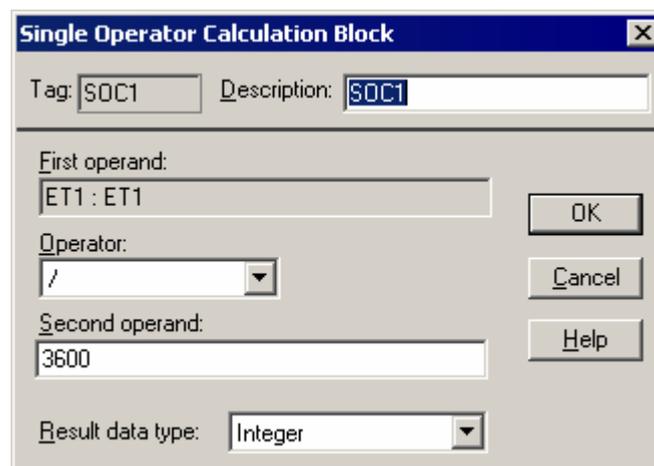


Рис. 4.148. Конфигурирование блока деления SOS1

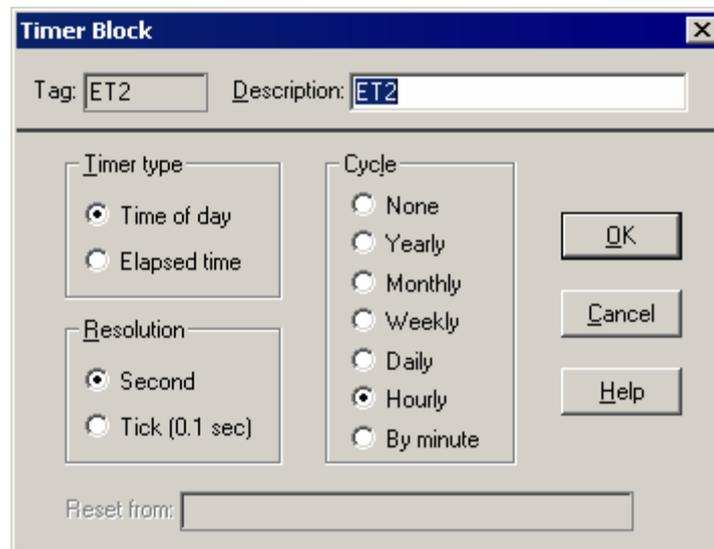


Рис. 4.149. Конфигурирование блока таймера ET2

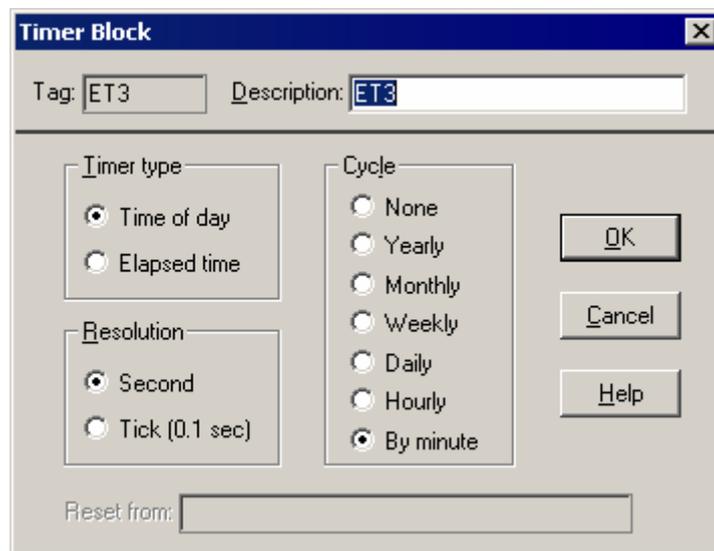


Рис. 4.150. Конфигурирование блока таймера ET3

Упражнение 4.35

Разработайте стратегию, обладающую следующими возможностями. Стратегия должна реализовать секундомер (секунды-десятые доли секунды). При нажатии

кнопки секундомер запускается, а при нажатии другой — фиксируется. Повторное нажатие кнопки сбрасывает показание, вновь запускает секундомер и т. д.

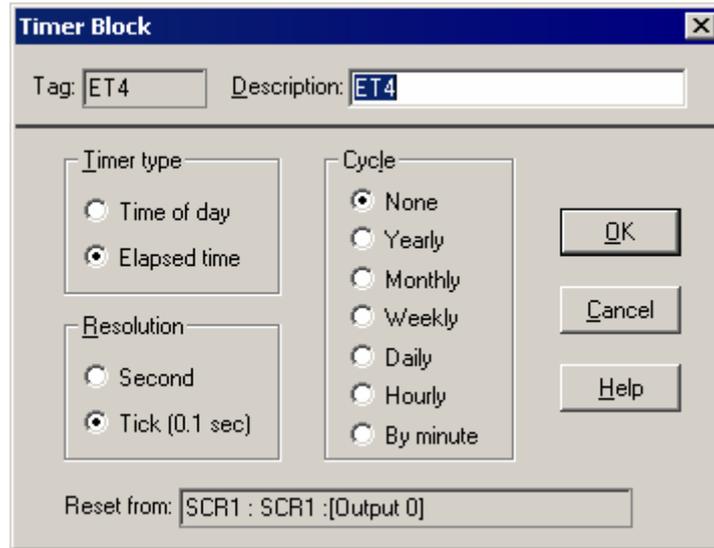


Рис. 4.151. Конфигурирование блока таймера ET4

Упражнение 4.36

Разработайте стратегию, обладающую следующими возможностями. Стратегия должна реализовать секундомер (секунды-десятые доли секунды). При нажатии кнопки секундомер запускается, а при ее отжати — фиксируется. Сброс показания секундомера — от отдельной кнопки.

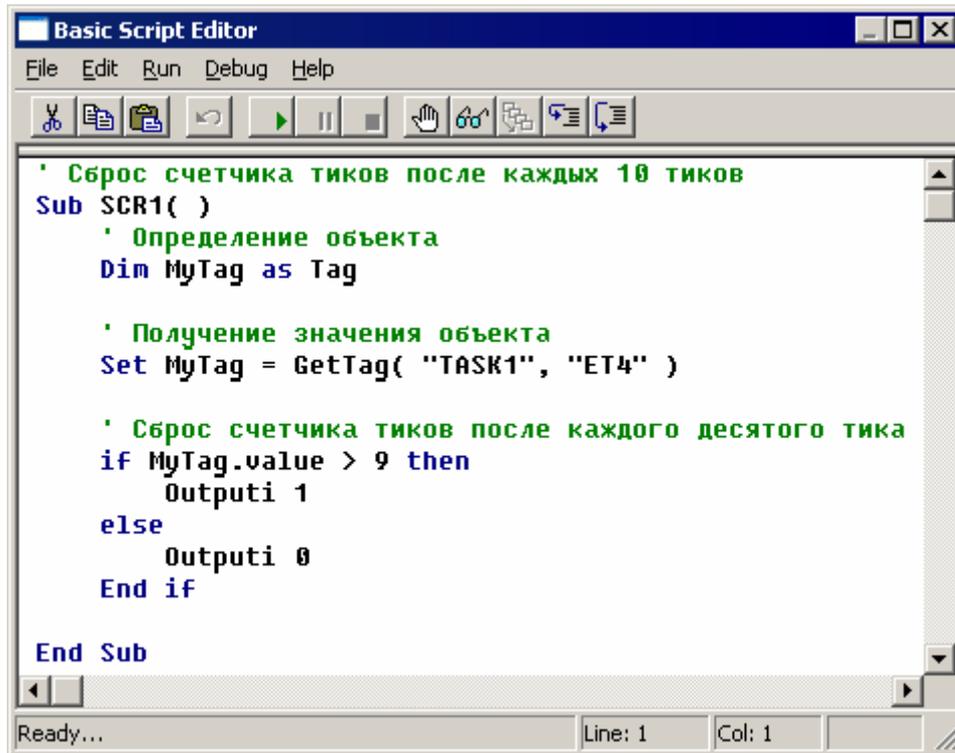
Упражнение 4.37

Разработайте стратегию, обладающую следующими возможностями. Стратегия должна реализовать секундомер (секунды-десятые доли секунды). При нажатии кнопки секундомер запускается, а при ее отжати — фиксируется. Использовать один цифровой индикатор.

Замечание

В следующей далее главе приводится методика информационной интеграции SCADA-системы GeniDAQ и промышленного контроллера, базирующаяся на использовании OPC-стандарта. В качестве промышленного контроллера используется контроллер типа 7188E2 фирмы ICP DAS (Тайвань), содержащий устройства связи с объектом (УСО), а в качестве OPC-сервера рассматривается OPC-сервер OpenLabOPCSvr.AiVT.2, разработанный на кафедре автоматике и вычислительной техники Санкт-Петербургского государственного политехнического университета с участием автора. Изложенная методика интересна тем, что может быть использована для информационной связи SCADA-системы GeniDAQ и имеющегося в распоряжении любого другого промышленного контроллера, снабженного OPC-сервером. Другой

пример информационной интеграции SCADA-системы **Vijeo Citect** (компаний Citect, Австралия и Schneider Electric, Франция) и промышленного контроллера **Twido** (Schneider Electric), снабженного групповым OPC-сервером **OPC Factory Server** (Schneider Electric), рассматривается далее во второй части учебного пособия.



```
Basic Script Editor
File Edit Run Debug Help
[Icons]
' Сброс счетчика тиков после каждых 10 тиков
Sub SCR1( )
' Определение объекта
Dim MyTag as Tag
' Получение значения объекта
Set MyTag = GetTag( "TASK1", "ET4" )
' Сброс счетчика тиков после каждого десятого тика
if MyTag.value > 9 then
    Outputi 1
else
    Outputi 0
End if
End Sub
Ready... Line: 1 Col: 1
```

Рис. 4.152. Конфигурирование блока Бейсик-сценария

Глава 5. OPC-технология информационной интеграции SCADA-системы и промышленного контроллера

В главе последовательно приводится краткое описание промышленного контроллера 7188E2 фирмы ICP DAS (Тайвань), рассматриваются интерфейс OPC-сервера OpenLabOPCSvr.AiVT.2 и его конфигурирование, настройка компьютеров локальной вычислительной сети и методика информационного обмена между функциональными блоками SCADA-системы и промышленного контроллера.

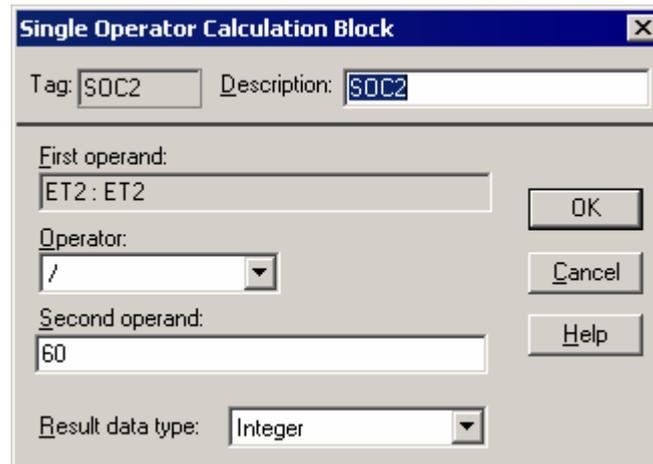


Рис. 4.153. Конфигурирование блока деления SOS2

5.1. Промышленный контроллер 7188E2 фирмы ICP DAS (Тайвань)

Промышленный контроллер типа 7188E2 фирмы ICP DAS (Тайвань) используется для взаимодействия с объектами управления и содержит устройства ввода-вывода аналоговой и дискретной информации. В настоящее время большинство новых моделей производимых контроллеров снабжено интерфейсным выходом на Ethernet. К подобным контроллерам обычно присоединяются пассивные модули ввода/вывода. Обращение из host-компьютера к модулям ввода/вывода происходит косвенно через контроллер. Именно такая конфигурация используется в рассматриваемом случае в качестве одного из возможных решений (рис. 5.1).

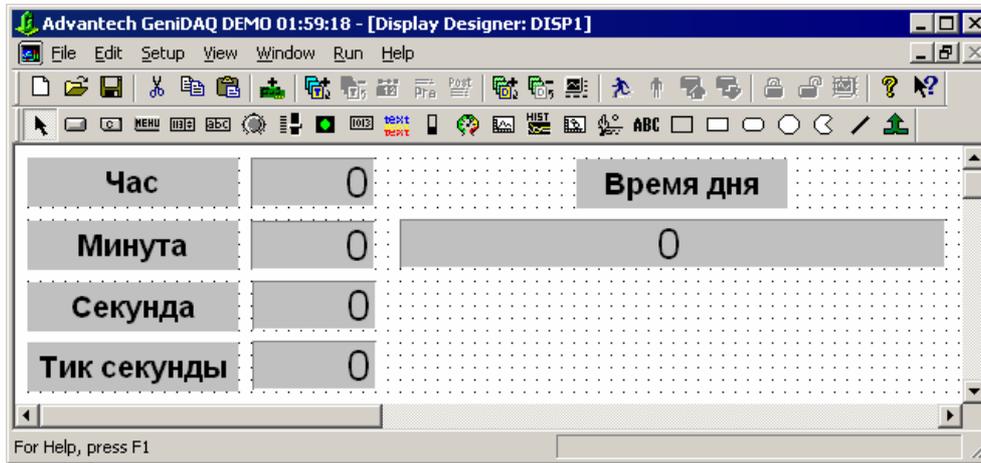


Рис. 4.154. Конфигурация окна отображения

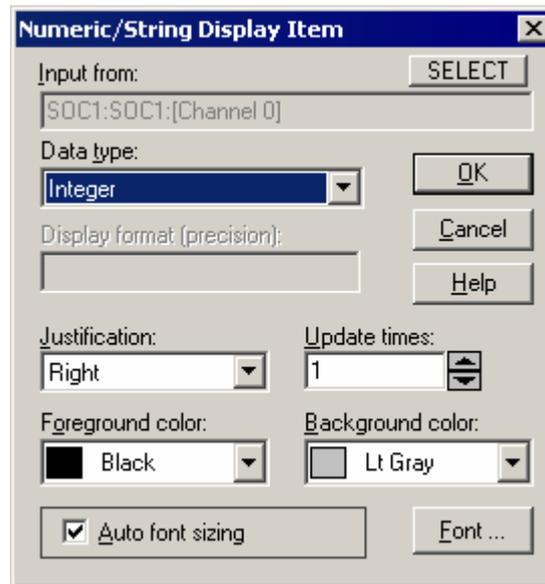


Рис. 4.155. Конфигурация цифрового индикатора для блока SOS1

Промышленный контроллер подсоединяется к host-компьютеру через выход Ethernet компьютера. В сети содержатся два восьмиканальных модуля аналогового ввода I-7017 (± 10 В), два четырехканальных модуля аналогового вывода I-7024 (± 10 В) и один модуль дискретного ввода/вывода I-7050 с семью входами, восемью выходами и семью счетчиками для дискретных входов.

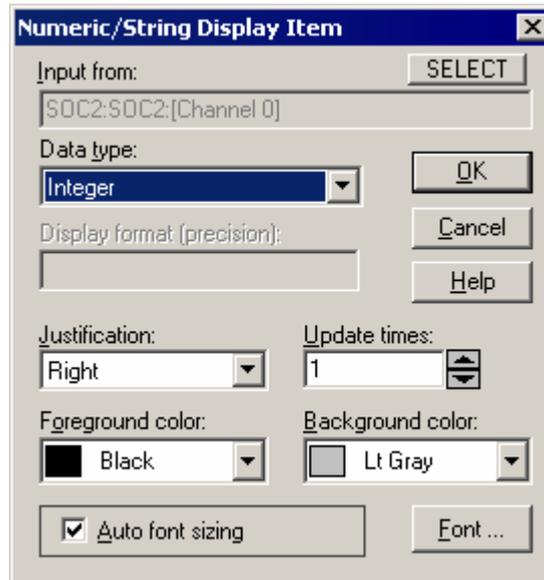


Рис. 4.156. Конфигурация цифрового индикатора для блока SOS2

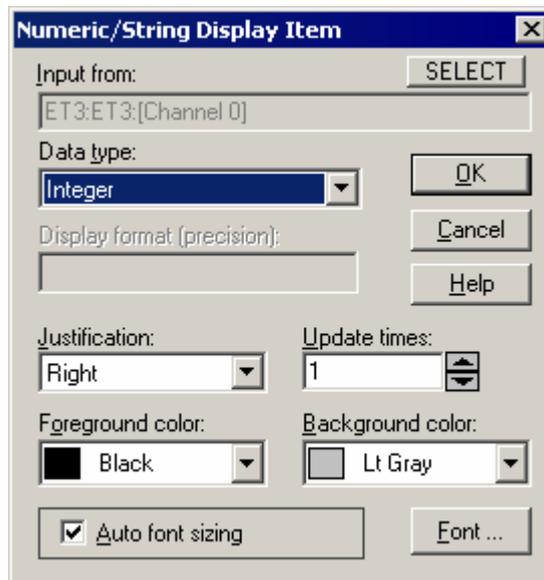


Рис. 4.157. Конфигурация цифрового индикатора для блока ET3

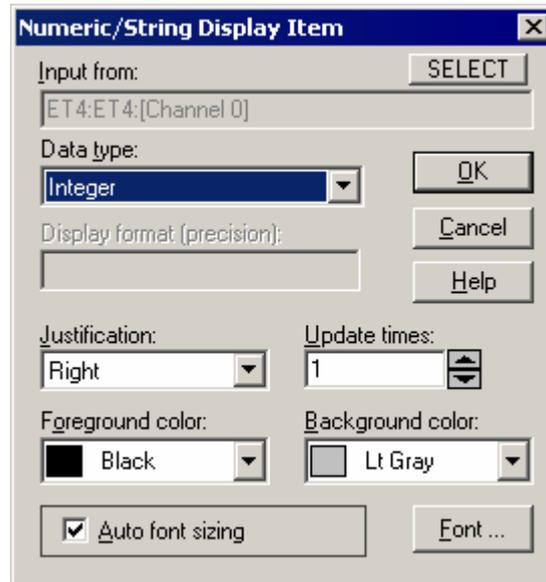


Рис. 4.158. Конфигурация цифрового индикатора для блока ET4

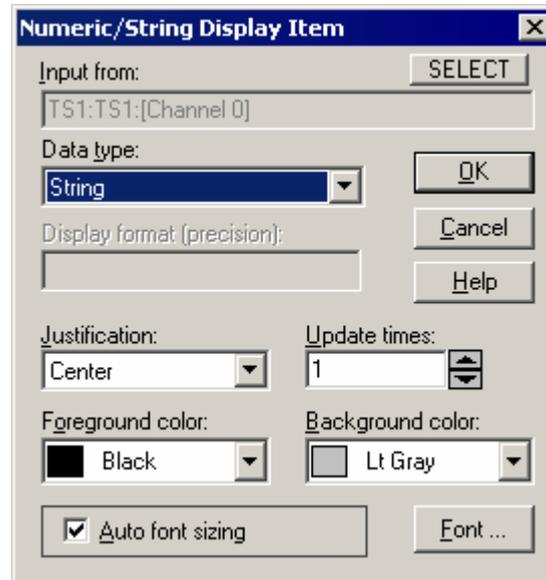


Рис. 4.159. Конфигурация цифрового индикатора для блока TS1

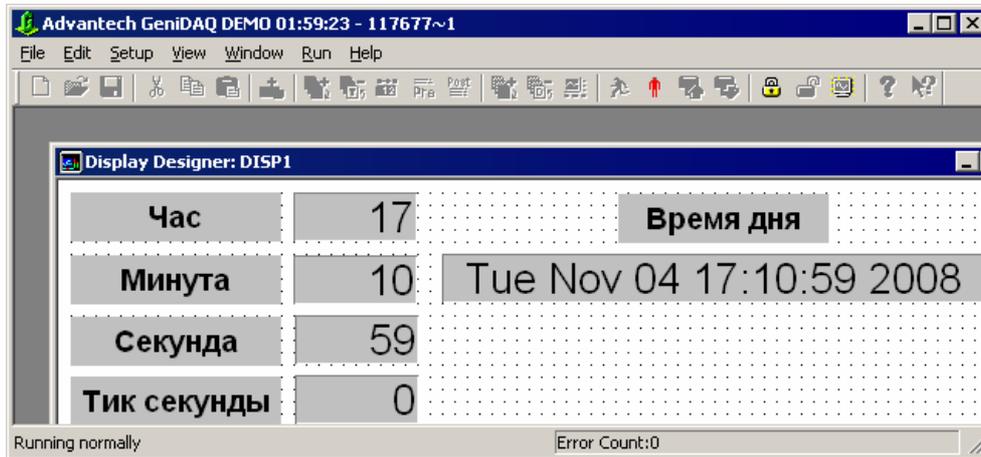


Рис. 4.160. Вид окна отображения при работе приложения

Каждый модуль имеет уникальный адрес (см. рис. 5.1). Для обеспечения доступа к контроллеру используется сервер VxComm, который создает в host-компьютере множество виртуальных COM-портов. Запросы ввода/вывода от OPC-сервера по интерфейсу RS-232 поступают в сервер VxComm и сервер, в свою очередь, преобразует сообщения RS-232 в пакеты TCP/IP и передает контроллеру 7188E2. При этом программисту нет необходимости программировать обмен по сети, например, знать протокол TCP/IP. Ему надо владеть лишь традиционным протоколом RS-232 и обладать навыками традиционного программирования в области промышленной автоматизации. Время доступа к данным контроллера и каналам ввода/вывода определяется скоростью передачи данных используемого виртуального COM-порта. Промышленный контроллер обладает следующими техническими характеристиками.

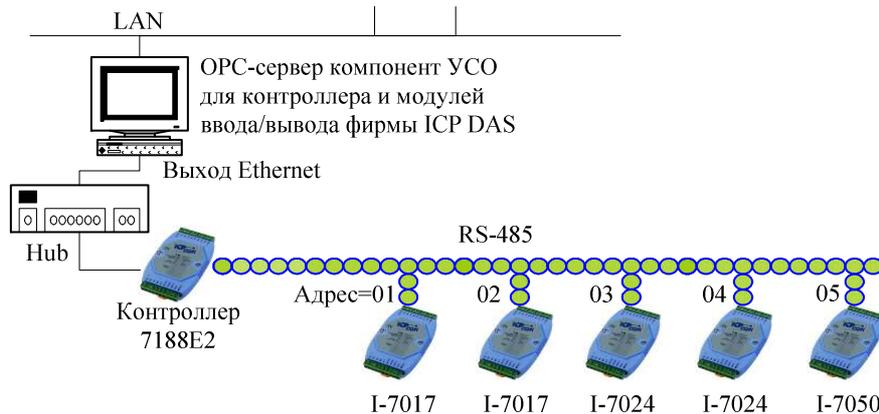


Рис. 5.1. Физическая структура системы мониторинга и управления модулями ввода-вывода промышленного контроллера типа 7188E2 фирмы ICP DAS

Процессор AMD80188ES с тактовой частотой 40 МГц. Операционная система реального времени MiniOS7, разработанная специально для контроллеров. Из нее исключены неиспользуемые для контроллеров функции, упрощена файловая система, добавлена поддержка модулей ввода-вывода и слотов расширения, устройств дополнительной памяти, функций самодиагностики, управления Flash-ROM, ускорена процедура загрузки и др. Flash-память и RAM-память по 256 Ком. Имеются встроенные часы реального времени и двойной сторожевой таймер.

Модуль ввода аналоговых сигналов I-7017 имеет восемь входных канала ± 10 Вольт, 16 разрядов, скорость опроса 10 сигналов в секунду. В составе контроллера используются два модуля.

Модуль вывода аналоговых сигналов I-7024 имеет четыре выходных канала ± 10 Вольт, 12 разрядов, скорость опроса 10 сигналов в секунду. В составе контроллера используются два модуля.

Модуль дискретного ввода-вывода I-7050 имеет семь входных канала для сигналов TTL-уровня (логический 0 соответствует диапазону 0-1 Вольт, логическая 1 – диапазону 3.5-30 Вольт), семь счетчиков для входных сигналов (входная частота до 100 Гц) и восемь выходных каналов с открытым коллектором (30 Вольт, 30 мА). Скорость опроса 10 сигналов в секунду. В составе контроллера используется один модуль.

5.2. Интерфейс OPC-сервера OpenLabOPCSvr.AiVT.2 и его конфигурирование

После включения компьютера, на котором реализован OPC-сервер, и промышленного контроллера на экране компьютера появляется запрос на ввод атрибутов пользователя (имя, пароль, права администратора), которые следует получить от преподавателя. Для запуска OPC-сервера с подключенным эмулятором тестовых сигналов достаточно запустить файл Системный диск\Program Files\OpenLabOPCSvr_7188E&Emulator\OpenLabOPCSvr.exe. Для этой цели удобнее всего воспользоваться ярлыком **7188E2&Emulator**, который следует предварительно разместить на рабочем столе. В результате этого на экране появляется окно, представленное на рис. 5.2.

В этом окне на переднем плане появляется диалоговое окно, информирующее о способе задания параметров эмулируемых сигналов. Нажмите в диалоговом окне кнопку **ОК**.

Для управления работой и конфигурирования OPC-сервера и эмулятора тестовых сигналов в окне, представленном на рис. 5.2, имеется строка меню и панель инструментов. По умолчанию, в левом поле окна выбран эмулятор тестовых сигналов **Emulator**, а в правом поле отображается информация о составе и параметрах тестовых сигналов. Если в левом поле выбрать контроллер **7188E2** и нажать кнопку **Monitoring** на панели инструментов, то в правом поле будет отображена информация об уже сконфигурированных тегах контроллера (рис. 5.3).

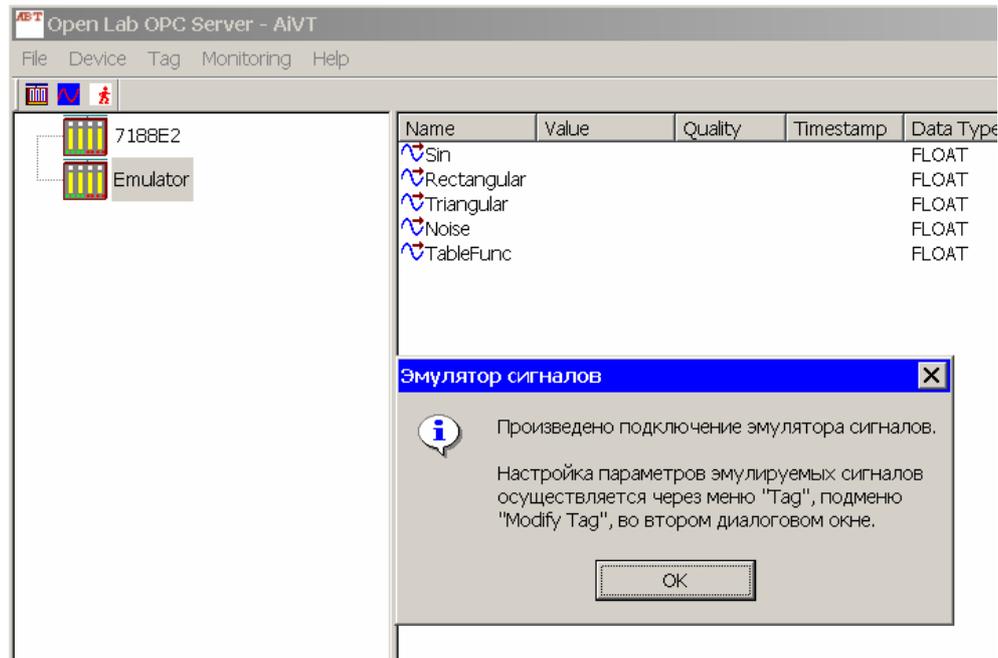


Рис. 5.2. Вид окна после запуска OPC-сервера **OpenLabOPCSvr.AiVT.2** с эмулятором тестовых сигналов

Интерфейс OPC-сервера с эмулятором представлен строкой меню с пятью меню, панелью инструментов с тремя кнопками, двумя названными ранее окнами и строкой состояния (статуса). Меню **File** содержит команды **Open Config** (акселератор **Ctrl+O**) — выбор и загрузка из файла ранее созданной конфигурации, **Save Config** (акселератор **Ctrl+S**) — сохранение в файле текущей конфигурации, **Exit** — завершение работы сервера. Меню **Device** содержит команды **Load New Device** (акселератор **Ctrl+L**, кнопка **New Device** на панели инструментов) — создание нового устройства, **Modify Device** (акселератор **Ctrl+M**) — модификация выбранного устройства, **Unload Device** (акселератор **Ctrl+U**) — удаление из конфигурации выбранного устройства. На приведенном ранее рис. 5.3 текущая конфигурация содержит устройства **7188E2** (контроллер) и **Emulator**. Меню **Tag** содержит команды **Add New Tag** (акселератор **Ctrl+A**, кнопка **New Tag** на панели инструментов) — добавление в устройство нового тега, **Modify Tag** (акселератор **Ctrl+F**) — модификация выбранного тега, **Delete Tag** (акселератор **Del**) — удаление из конфигурации выбранного тега. С помощью команды **Tag | Modify Tag** можно модифицировать тестовый сигнал эмулятора. Например, если выбрать тестовый сигнал **Sin**, и выполнить для него команду **Tag | Modify Tag** и нажать кнопку **OK**, то появится диалоговое окно, представленное на рис. 5.4.

Name	Value	Quality	Timestamp	Data Type	Signal Type	Units	Source Tag	Description
7017.1.0	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.0	ABK 1-1
7017.1.1	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.1	ABK 1-2
7017.1.2	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.2	ABK 1-3
7017.1.3	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.3	ABK 2-1
7017.1.4	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.4	ABK 2-2
7017.1.5	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.5	ABK 2-3
7017.1.6	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.6	ABK 3-1
7017.1.7	-0.03	GOOD	11:34:16	FLOAT	Input		7017.1.7	ABK 3-2
7017.2.0	-0.03	GOOD	11:34:16	FLOAT	Input		7017.2.0	ABK 3-3
7017.2.1	-0.02	GOOD	11:34:16	FLOAT	Input		7017.2.1	ABK 4-1
7017.2.2	-0.02	GOOD	11:34:16	FLOAT	Input		7017.2.2	ABK 4-2
7017.2.3	-0.02	GOOD	11:34:16	FLOAT	Input		7017.2.3	ABK 4-3
7017.2.4	-0.03	GOOD	11:34:16	FLOAT	Input		7017.2.4	ABK 5-1
7017.2.5	-0.02	GOOD	11:34:16	FLOAT	Input		7017.2.5	ABK 5-2
7017.2.6	-0.02	GOOD	11:34:16	FLOAT	Input		7017.2.6	ABK 5-3
7017.2.7	-0.02	GOOD	11:34:16	FLOAT	Input		7017.2.7	ABK 6-1
7024.3.0	0.00	BAD	03:00:00	FLOAT	Output		7024.3.0	ABK 1-6
7024.3.1	0.00	BAD	03:00:00	FLOAT	Output		7024.3.1	ABK 2-6
7024.3.2	0.00	BAD	03:00:00	FLOAT	Output		7024.3.2	ABK 3-6
7024.3.3	0.00	BAD	03:00:00	FLOAT	Output		7024.3.3	ABK 4-6
7024.4.0	0.00	BAD	03:00:00	FLOAT	Output		7024.4.0	ABK 5-6
7024.4.1	0.00	BAD	03:00:00	FLOAT	Output		7024.4.1	ABK 6-6
7024.4.2	0.00	UNCERT...	03:00:00	FLOAT	Output		7024.4.2	ABK 1-4
7024.4.3	0.00	BAD	03:00:00	FLOAT	Output		7024.4.3	ABK 2-4
7050.5.DI.0	OFF	GOOD	11:34:17	BOOL	Input		7050.5.DI.0	X0
7050.5.DI.1	OFF	GOOD	11:34:17	BOOL	Input		7050.5.DI.1	X1
7050.5.DI.2	OFF	GOOD	11:34:17	BOOL	Input		7050.5.DI.2	X2
7050.5.DI.3	OFF	GOOD	11:34:17	BOOL	Input		7050.5.DI.3	X3
7050.5.DI.4	ON	GOOD	11:34:17	BOOL	Input		7050.5.DI.4	X8
7050.5.DI.5	ON	GOOD	11:34:17	BOOL	Input		7050.5.DI.5	X9
7050.5.DI.6	ON	GOOD	11:34:17	BOOL	Input		7050.5.DI.6	X10
7050.5.DI.Counter.0	0.00	GOOD	11:34:17	FLOAT	Input		7050.5.DI.Counter.0	
7050.5.DI.Counter.1	0.00	GOOD	11:34:17	FLOAT	Input		7050.5.DI.Counter.1	
7050.5.DI.Counter.2	0.00	GOOD	11:34:17	FLOAT	Input		7050.5.DI.Counter.2	
7050.5.DI.Counter.3	0.00	GOOD	11:34:17	FLOAT	Input		7050.5.DI.Counter.3	
7050.5.DI.Counter.4	0.00	GOOD	11:34:18	FLOAT	Input		7050.5.DI.Counter.4	
7050.5.DI.Counter.5	0.00	GOOD	11:34:18	FLOAT	Input		7050.5.DI.Counter.5	
7050.5.DI.Counter.6	0.00	GOOD	11:34:18	FLOAT	Input		7050.5.DI.Counter.6	
7050.5.DO.0	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.0	Y0
7050.5.DO.1	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.1	Y1
7050.5.DO.2	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.2	Y2
7050.5.DO.3	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.3	Y3
7050.5.DO.4	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.4	Y4
7050.5.DO.5	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.5	Y5
7050.5.DO.6	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.6	Y6
7050.5.DO.7	OFF	BAD	03:00:00	BOOL	Output		7050.5.DO.7	Y7

Рис. 5.3. Мониторинг сконфигурированных тегов контроллера

В этом окне можно посмотреть текущие значения параметров тестового сигнала или изменить их. Для выхода из диалогового окна следует нажать кнопку **ОК**. Меню **Monitoring** содержит команды **Toolbar** — отображение или скрытие панели инструментов, **Status Bar** — отображение или скрытие строки состояния (статуса), **Tag Monitoring** (кнопка **Monitoring**) — мониторинг тегов выбранного устройства. И, наконец, команда **Help | About OPC Server...** с помощью диалогового окна отображает информацию об OPC-сервере.

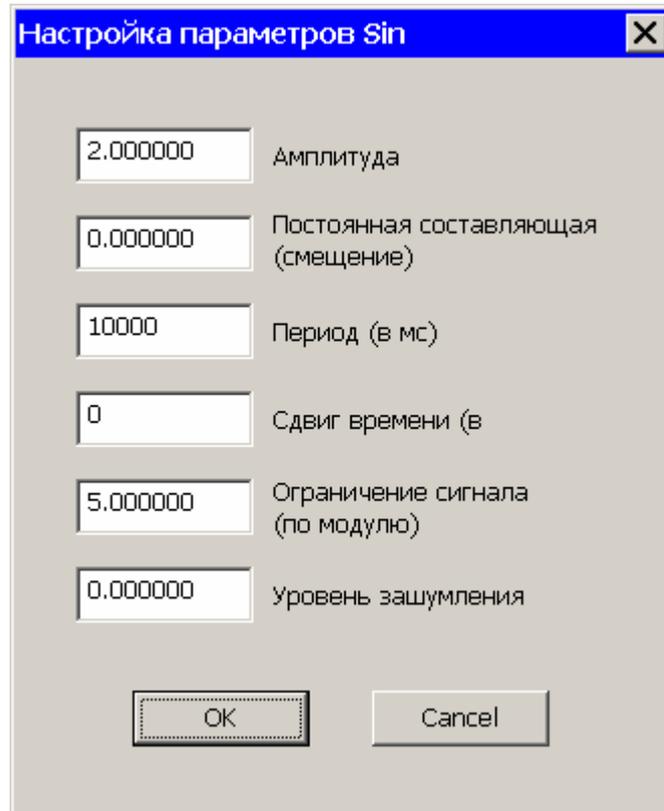


Рис. 5.4. Модификация тестового сигнала Sin эмулятора

5.3. Настройка компьютеров локальной вычислительной сети

Для обеспечения доступа к OPC-серверу из любого компьютера локальной вычислительной сети следует выполнить дополнительную настройку компьютеров, которая несколько отличается для различных операционных систем.

5.3.1. Настройка ОС Windows2000 для работы с OPC-серверами (на примере OPC-сервера OpenLabOPCSvr.AiVT.2)

1. При регистрации входа на компьютере *обязательно* используйте имя пользователя и пароль как у компьютера, на котором работает OPC-сервер, и с правами администратора.

2. В папке **..\Program Files** создайте папку **OpenLabOPCSvr** (можно использовать и другое имя папки, но лучше, чтобы имя папки совпадало с именем OPC-сервера). Из ЭВМ, где работает OPC-сервер (пусть, например, такой компьютер имеет имя **Comp1_2003**) из аналогичной папки в созданную папку копируем файлы **OPCProxy.dll** и **Setup.reg**.
3. Запустите на выполнение скопированный файл **Setup.reg**. В появляющихся дважды диалоговых окнах нажмите кнопку **ОК**.
4. Зарегистрируйте **OPCProxy**, для чего в командной строке выполните команду

```
regsvr32 OPCProxy.dll
```

В появившемся диалоговом окне нажмите кнопку **ОК**.
5. Сконфигурируйте DCOM. С этой целью выполните команду **Пуск | Выполнить...**, в появившемся диалоговом окне (рис. 5.5) введите **dcomcnfg** и нажмите кнопку **ОК**. В ответ на появившийся запрос нажмите кнопку **Да**.

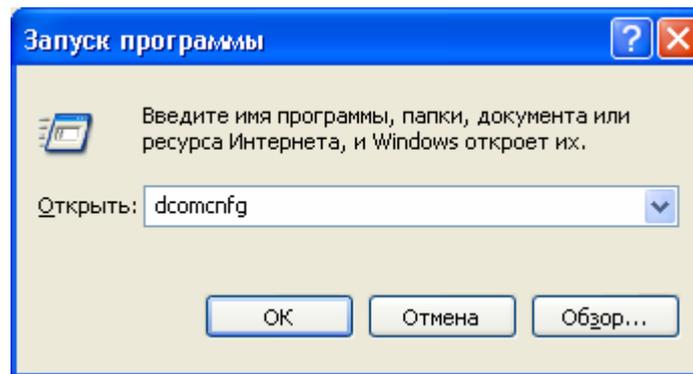


Рис. 5.5. Конфигурирование DCOM

6. В появившемся диалоговом окне (рис. 5.6) выберите OPC-сервер (в нашем примере это **Opened Lab OPC Server**) и нажмите кнопку **Свойства...**
7. В появившемся диалоговом окне выберите вкладку **Свойства** и настройте ее в соответствии с рис. 5.7. Здесь, в качестве примера, **Comp1_2003** является именем компьютера, на котором работает OPC-сервер. После настройки дважды нажмите кнопку **ОК**.

5.3.2. Настройка ОС Windows XP для работы с OPC-серверами (на примере OPC-сервера OpenLabOPCSvr.AiVT.2)

1. При регистрации входа на компьютере *обязательно* используйте имя пользователя и пароль как у компьютера, на котором работает OPC-сервер, и с правами администратора.

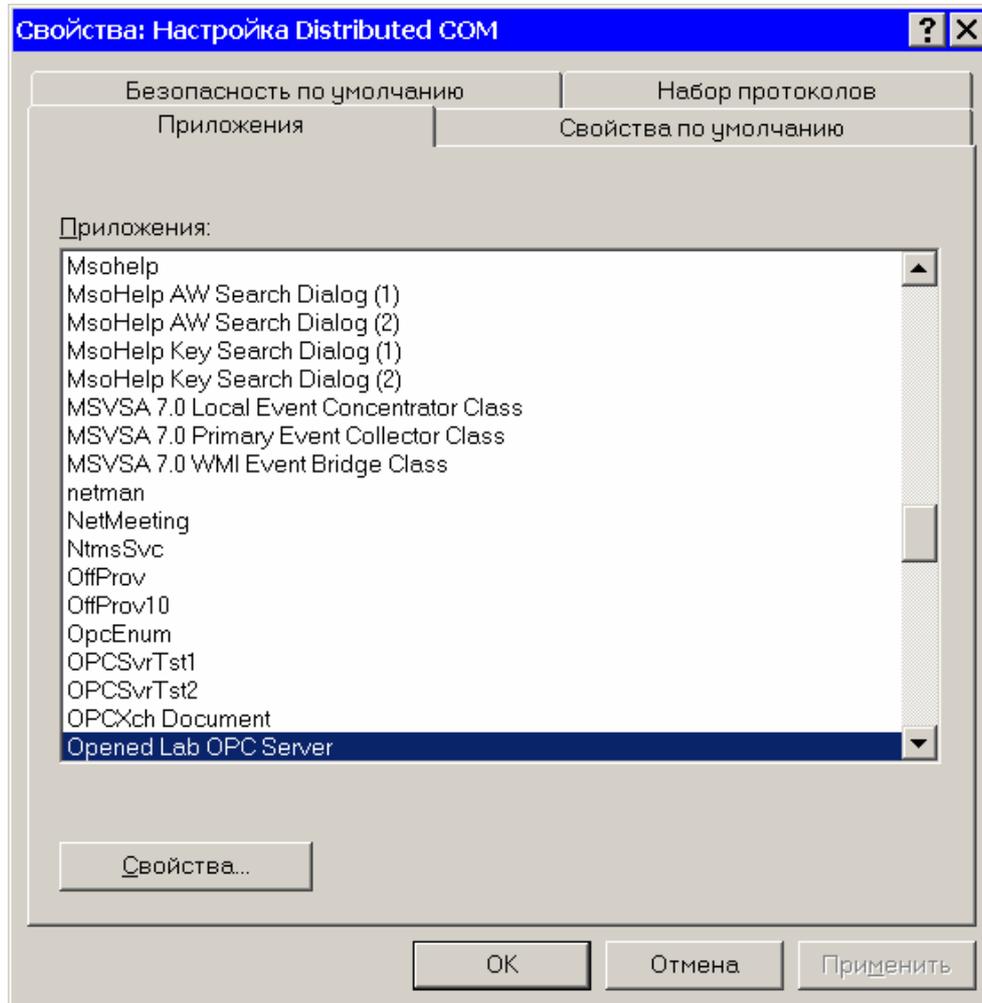


Рис. 5.6. Выбор OPC-сервера

2. В папке **..\Program Files** создайте папку **OpenLabOPCSvr** (можно использовать и другое имя папки, но лучше, чтобы имя папки совпадало с именем OPC-сервера). Из ЭВМ, где работает OPC-сервер (пусть, например, такой компьютер имеет имя **Comp1_2003**) из аналогичной папки в созданную папку копируем файлы **OPCProxy.dll** и **Setup.reg**.
3. Запустите на выполнение скопированный файл **Setup.reg**. В появляющихся дважды диалоговых окнах нажмите кнопку **ОК**.
4. Зарегистрируйте **OPCProxy**, для чего в командной строке выполните команду
`regsvr32 OPCProxy.dll`

В появившемся диалоговом окне нажмите кнопку **ОК**.

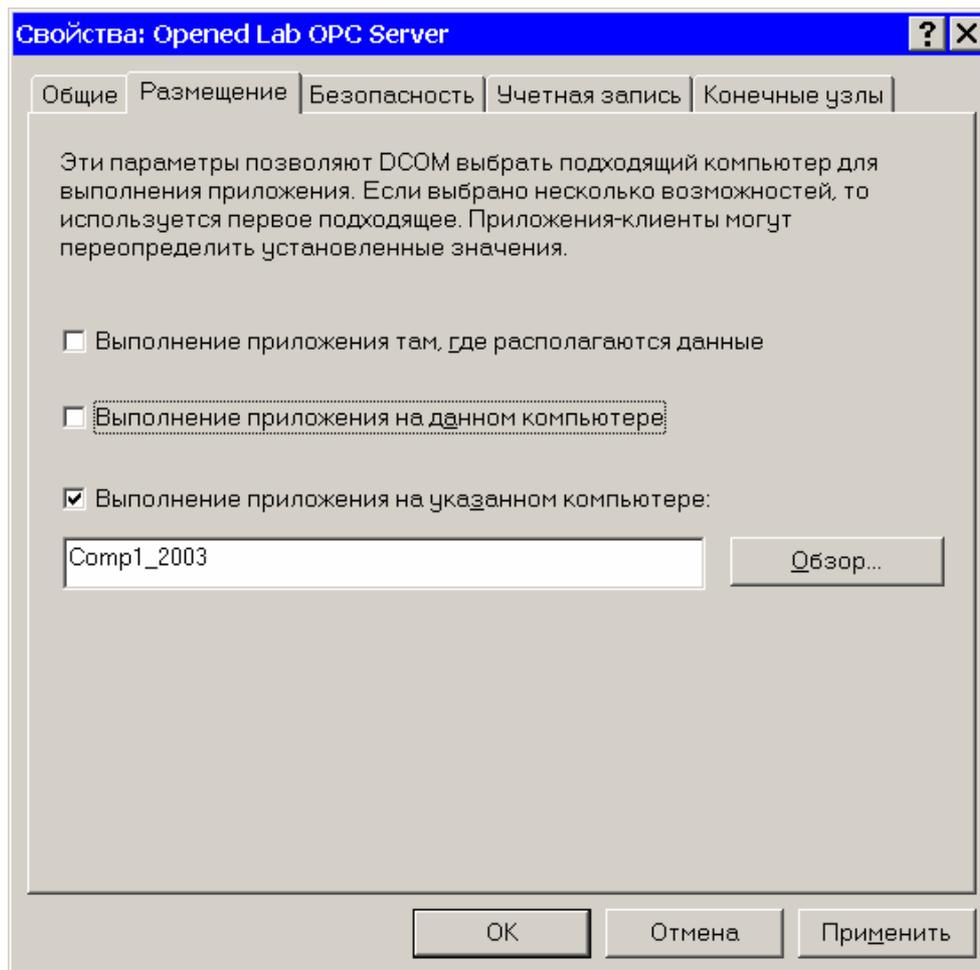


Рис. 5.7. Указание размещения OPC-сервера

5. Для выбранного сервера правой кнопкой мыши вызовите контекстное меню, выполните команду **Свойства**, в появившемся окне **Свойства** выберите вкладку **Расположение** и настройте ее в соответствии с рис. 5.9. Здесь, в качестве примера, **Comp1_2003** является именем компьютера, на котором работает OPC-сервер. После настройки дважды нажмите кнопку **ОК**.
6. Сконфигурируйте DCOM. С этой целью выполните команду **Пуск | Выполнить...**, в появившемся диалоговом окне (см. приведенный ранее рис. 5.5) введите **dcomcnfg** и нажмите кнопку **ОК**.

7. В появившемся диалоговом окне **Службы компонентов** откройте последовательно кнопки **Службы компонентов**, **Компьютеры**, **Мой компьютер**, **Настройка DCOM** и выберите OPC-сервер (рис. 5.8, в нашем примере это **Opened Lab OPC Server**).

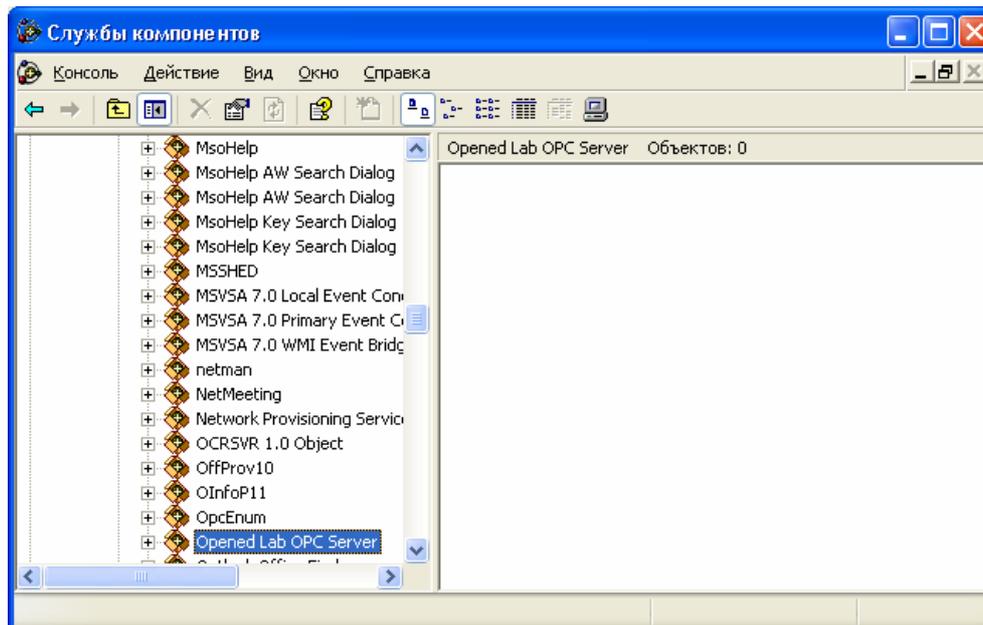


Рис. 5.8. Выбор OPC-сервера

8. Установите низкий уровень безопасности, для чего выполните команду **Пуск | Настройка | Панель управления | Брандмауэр Windows | Вкладка Общие** и выберите радиокнопку **Выключить**.

5.4. Занятие 12 "Информационный обмен между SCADA-системой и промышленным контроллером"

Цель занятия состоит в демонстрации средств, предоставляемых GeniDAQ, для ввода-вывода аналоговой и дискретной информации в реальном масштабе времени и ее отображения. В отличие от занятия 1 вместо программного эмулятора сигналов **Advantech I/O** используется OPC-сервер (**OpenLabOPCSvr.AiVT.2**), блоки аналогового и дискретного ввода-вывода и различные индикаторы.

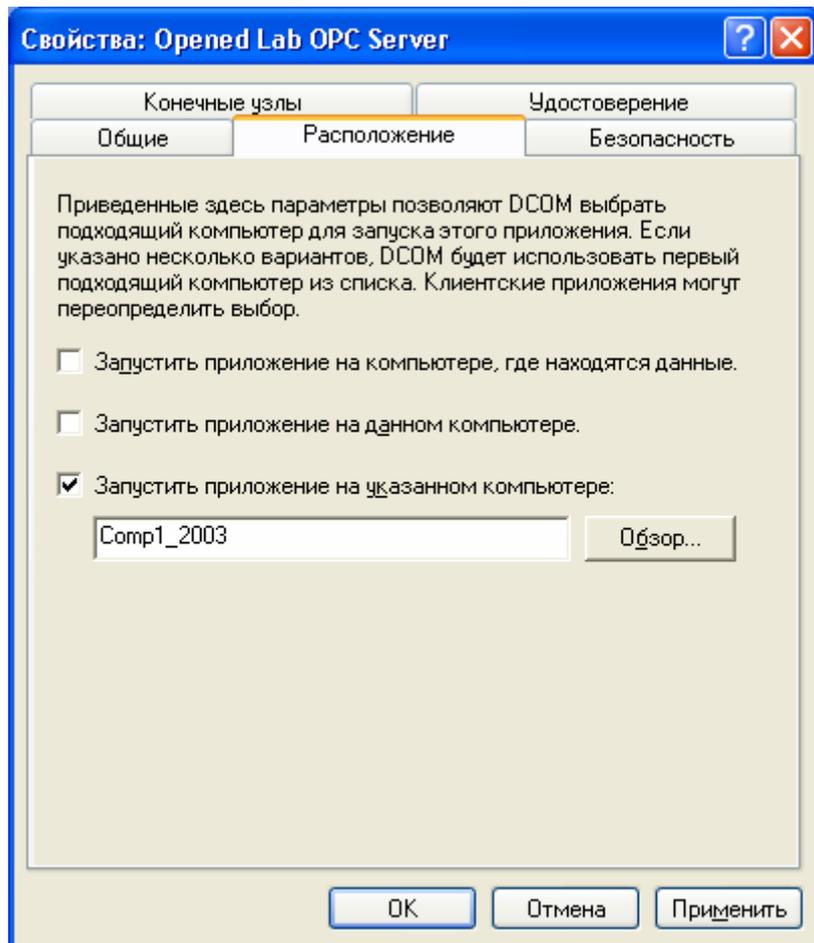


Рис. 5.9. Указание размещения OPC-сервера

5.4.1. Используемый инструментарий

Новыми инструментами, используемыми на данном занятии, являются блок аналогового вывода (редактор задач); блок дискретного ввода (редактор задач); блок дискретного вывода (редактор задач); OPC-сервер (рассмотрен ранее в начале главы); промышленный контроллер типа 7188E2 фирмы ICP DAS, содержащий УСО (рассмотрен ранее в начале главы); аналоговые вычислительные комплексы АВК-31, используемые для имитации аналоговых сигналов и моделирования объектов управления; имитатор для получения входных дискретных сигналов и приема выходных дискретных сигналов.

Блок аналогового вывода предназначен для передачи информации, получаемой от других функциональных блоков, элементов отображения или других приложений

Windows с использованием технологии COM/DCOM, устройствам, имеющим подсистему вывода аналоговых сигналов. Двойной щелчок левой клавишей мыши на пиктограмме **АО** блока аналогового вывода приводит к появлению диалогового окна для настройки параметров блока (рис. 5.10). Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров блока аналогового вывода будут рассмотрены далее.

Блок дискретного ввода предназначен для приема информации, в том числе с использованием технологии COM/DCOM, от устройств, имеющих подсистему ввода дискретных сигналов, и передачи указанных сигналов другим функциональным блокам и элементам отображения. Двойной щелчок левой клавишей мыши на пиктограмме **ДИ** блока дискретного ввода приводит к появлению окна диалогового окна для настройки параметров блока (рис. 5.11). Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров блока дискретного ввода будут рассмотрены далее.

Блок дискретного вывода предназначен для передачи информации, получаемой от других функциональных блоков, элементов отображения или других приложений Windows с использованием технологии COM/DCOM, устройствам, имеющим подсистему вывода дискретных сигналов. Двойной щелчок левой клавишей мыши на пиктограмме **ДО** блока дискретного вывода приводит к появлению диалогового окна для настройки параметров блока (рис. 5.12). Информацию о параметрах настройки этого окна можно получить с помощью кнопки **Help**. Конкретные настройки параметров блока дискретного вывода будут рассмотрены далее.

Соединитель блоков (проводник). В учебных примерах, рассмотренных ранее, вы уже неоднократно пользовались проводником. Соберем воедино все, что мы знаем об этом блоке. Пиктограмма этого инструмента, предназначенного для установления видимых связей между пиктограммами функциональных блоков задачи, расположена второй слева на панели инструментов редактора задач (см. приведенный ранее рис. 4.1). При выборе данного инструмента активизируется режим установления связей между функциональными блоками редактора задач и курсор мыши принимает вид катушки с нитками. Соединение функциональных блоков в рамках задачи является основным типом организации взаимодействия между блоками. Для соединения двух блоков следует поместить курсор в виде катушки с нитками на пиктограмму блока, данные которого предполагается передать в другой блок, и щелкнуть левой клавишей мыши.

Если блок, являющийся источником данных для второго блока, имеет несколько выходов, то в появившейся диалоговой панели со списком всех выходов блока потребуется выбрать номер соединяемого выхода. Далее необходимо поместить курсор на пиктограмму функционального блока, являющегося приемником данных и произвести щелчок левой клавишей мыши.

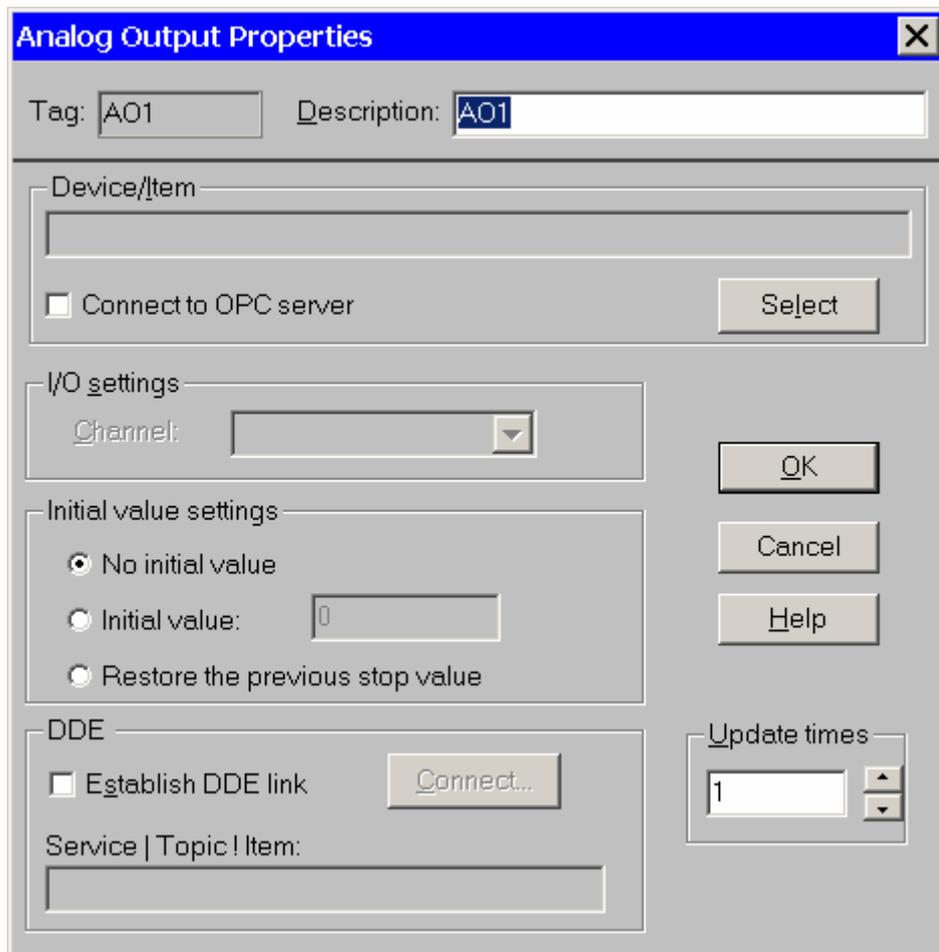


Рис. 5.10. Диалоговое окно настройки блока аналогового вывода

Если блок, являющийся приемником данных, имеет несколько входов, то в появившейся диалоговой панели со списком всех входов блока потребуется выбрать номер соединяемого входа. Если графическое изображение устанавливаемого соединения необходимо выполнить в виде ломаной линии, то для фиксации каждого отрезка ломаной следует производить щелчок левой клавишей мыши в точке, которая должна служить началом очередного отрезка ломаной. В зависимости от того, на каком блоке был произведен первый щелчок левой клавишей мыши при установлении связи, информационный поток через установленную логическую связь будет различным. Например, если требуется установить связь между блоком аналогового ввода и блоком усреднения, следует сначала произвести щелчок левой клавишей мыши на пиктограмме блока аналогового ввода, а затем на пиктограмме блока усреднения.

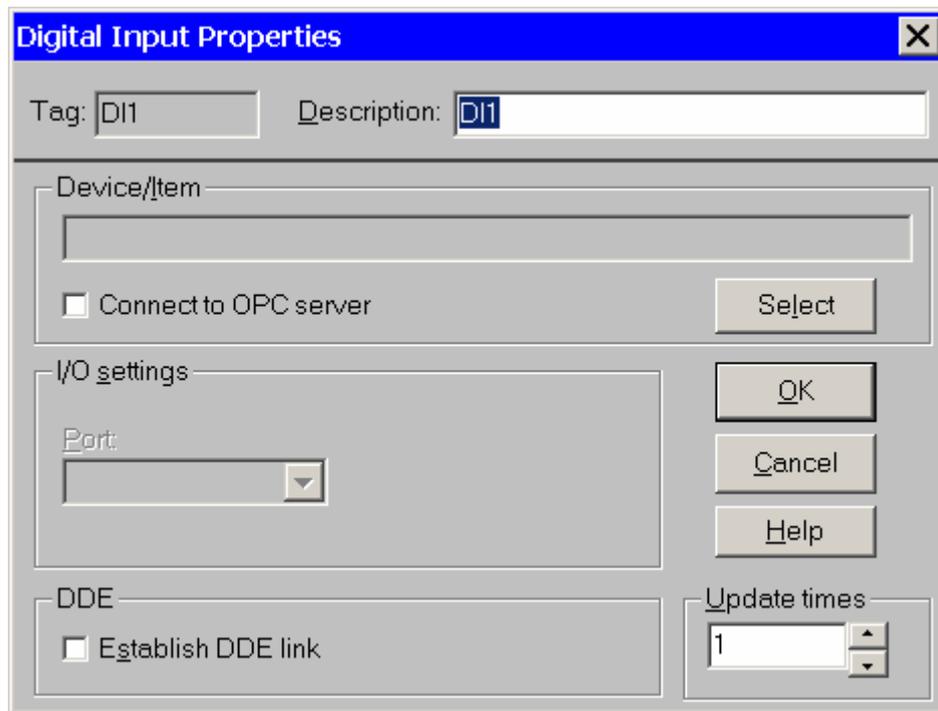


Рис. 5.11. Диалоговое окно настройки блока дискретного ввода

При этом место присоединения проводника к пиктограмме блока усреднения будет отмечено стрелкой, направленной в сторону блока усреднения. Направление стрелки указывает, что значение с выхода функционального блока аналогового ввода поступает на вход блока усреднения. Некоторые блоки допускают передачу данных только в одном направлении. Обратите внимание, что при попытке установления связи между блоком аналогового ввода и блоком ввода данных из файла на экран монитора будет выведено сообщение "Вход недоступен". Блок ввода данных из файла имеет только выходные каналы и не позволяет принимать данные.

Аналоговые вычислительные комплексы АВК-31 используются для имитации аналоговых входов, приема аналоговой информации и моделирования объектов управления. В каждом из АВК по три гнезда на коммутационном поле с маркировкой **Внешние цепи** (1, 2, 3) связаны с входами модулей ввода аналоговых сигналов промышленного контроллера. Аналогичным образом, выходы модулей аналогового выхода промышленного контроллера соединены с гнездами 4 и 6 на коммутационном поле АВК.

Имитатор для получения входных дискретных сигналов и приема выходных дискретных сигналов содержит по восемь тумблеров (X0-X7) и кнопок (X8-X15) для имитации входных дискретных сигналов и восемь светодиодов для индикации выдаваемых дискретных сигналов.

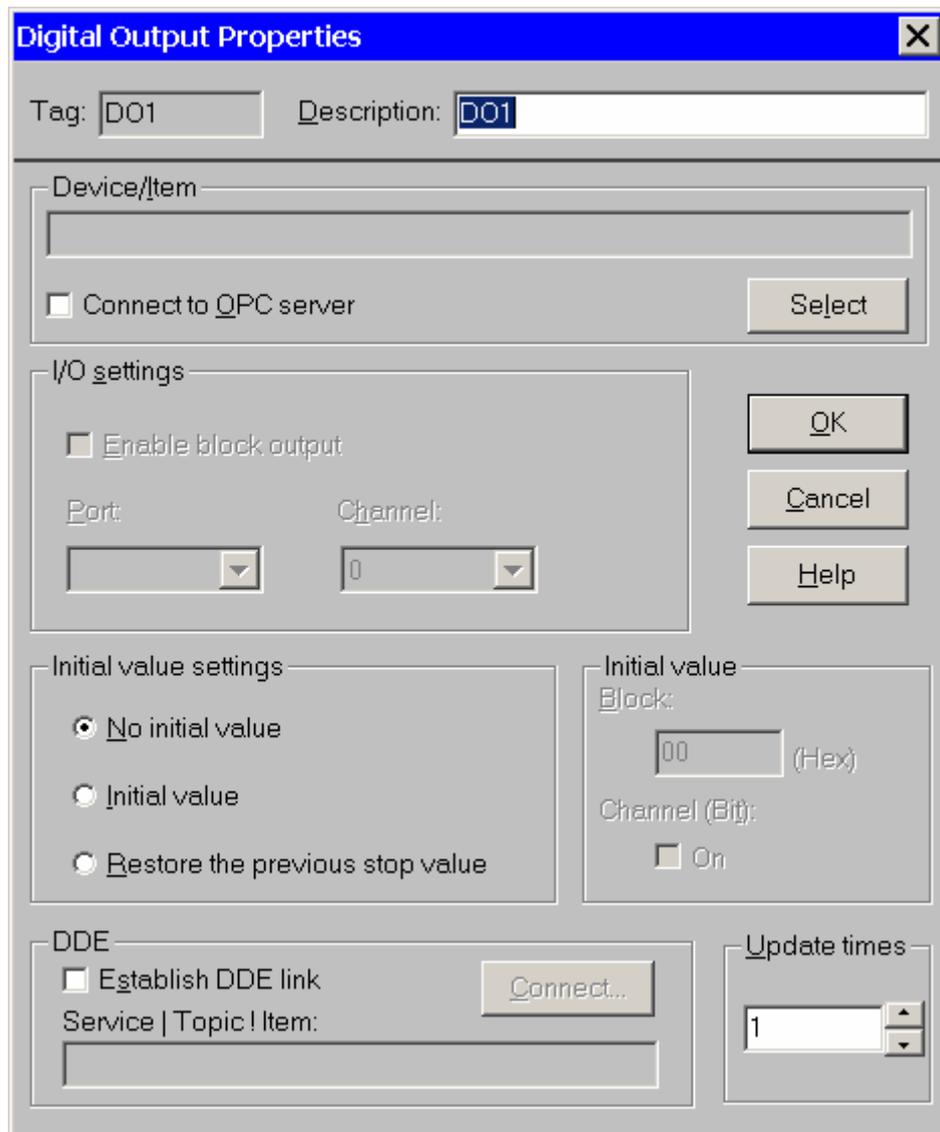


Рис. 5.12. Диалоговое окно настройки блока дискретного вывода

Имитатор для получения входных дискретных сигналов и приема выходных дискретных сигналов содержит по восемь тумблеров (X0-X7) и кнопок (X8-X15) для имитации входных дискретных сигналов и восемь светодиодов для индикации выдаваемых дискретных сигналов.

5.4.2. Проектирование приложения

1. *Первое действие* — включение OPC-сервера и промышленного контроллера — выполняет преподаватель (см. приведенный ранее подразд. 5.2).
2. *Второе действие.* Включите рабочую станцию или перегрузите ее, при регистрации обязательно используйте *те же самые имя и пароль*, которые использовались при регистрации на компьютере С запущенным OPC-сервером. Запустите построитель приложений и настройте приложение аналогично рассмотренному ранее. Назовите приложение **Занятие 12**.
3. *Третье действие.* Задайте временные параметры запуска приложения аналогично тому, как это делалось на занятии 1. В нашем примере задан период повторения задачи 100 миллисекунд.
4. *Четвертое действие.* В окне редактора задач на панели инструментов выберите кнопку (пиктограмму) с мнемоникой **AI** и перетащите ее в окно редактора задач. При этом в окне редактора задач появится блок аналогового ввода с позиционным обозначением **AI1** (см. занятие 1, рис. 4.9). Произведите двойной щелчок левой кнопкой мыши над блоком аналогового ввода и появится диалоговое окно его настройки. Введите в этом окне в поле **Description:** значение **AI61** (последние две цифры означают следующее: 6 — номер рабочей станции, возможный диапазон номеров 1...6, а 1 — номер аналогового входа на рабочей станции, возможный диапазон номеров 1...3). Для выбора источника аналогового сигнала — OPC-сервера — следует включить флажок **Connect to OPC Server**. Далее нажмите кнопку **Select**, в появившемся окне диалога **OPC Item Select** выберите сервер **OpenLabOPCSrv.AiVT.2**, нажмите кнопку **Connect...**, откройте **OpenLabOPCSrv.AiVT.2** и **7188E2**, выберите **7017.2.7** и нажмите кнопку **OK** (рис. 5.13). Тем самым, для подключения аналогового сигнала было выбрано гнездо 1 **Внешние цепи** аналогового вычислительного комплекса (АВК) рабочей станции №6 (см. приведенный ранее рис. 5.3). Диалоговое окно конфигурирования блока аналогового ввода приобретает вид, представленный на рис. 5.14. Для сохранения значений параметров блока аналогового ввода нажмите кнопку **OK**.
5. *Пятое действие.* Аналогичным образом поместите в окно редактора задач блок аналогового вывода. Для этого на панели инструментов выберите кнопку с мнемоникой **AO** и перетащите ее в окно редактора задач. При этом в окне редактора задач появится блок аналогового вывода с позиционным обозначением **AO1**. Произведите двойной щелчок левой кнопкой мыши над блоком вывода аналоговой информации и появится диалоговое окно его настройки. Введите в этом окне **Description:** значение **A066** (последние две цифры означают следующее: 6 — номер рабочей станции, возможный диапазон номеров 1...6, а 6 — номер аналогового выхода на рабочей станции, возможный диапазон номеров 4 и 6). Для выбора источника аналогового сигнала — OPC-сервера — следует включить флажок **Connect to OPC Server**. Далее нажмите кнопку **Select**, в появившемся окне диалога **OPC Item Select** выберите сервер **OpenLabOPCSrv.AiVT.2**, нажмите кнопку **Connect...**, откройте

OpenLabOPCSrv.AiVT.2 и **7188E2**, выберите **7024.4.1** и нажмите кнопку **ОК** (рис. 5.15). Тем самым, для подключения аналогового сигнала было выбрано гнездо 6 **Внешние цепи** аналогового вычислительного комплекса (АВК) рабочей станции №6 (см. приведенный ранее рис. 5.3). Диалоговое окно конфигурирования блока аналогового ввода приобретает вид, представленный на рис. 5.16. Для сохранения значений параметров блока аналогового ввода нажмите кнопку **ОК**.

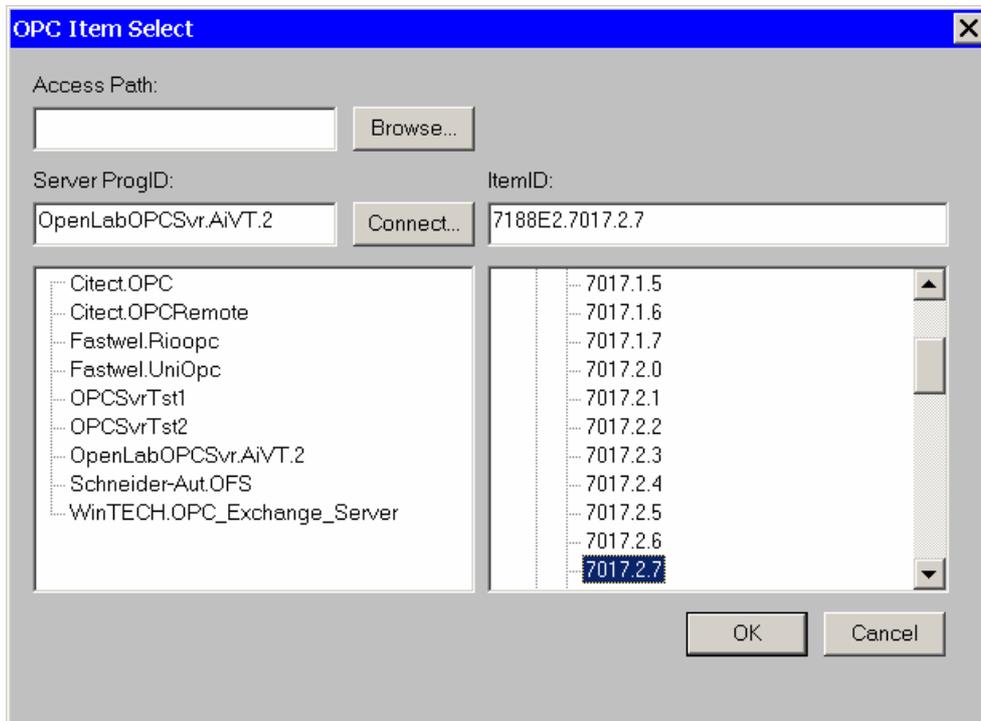


Рис. 5.13. Диалоговое окно выбора параметров OPC-сервера

6. *Шестое действие.* Теперь выполните соединение выхода блока аналогового ввода с входом блока аналогового вывода. Для этого выберите кнопку **Connection wire** на панели инструментов и поместите "нить" на блок **AI61** в окне редактора задач. В появившемся диалоговом окне выберите выходной канал **Output 0** (при работе с OPC-сервером правильным всегда будет только такой выбор), нажмите кнопку **ОК** и положите "нить" на блок **AO66** в окне редактора задач. В результате выполненных действий информация будет передаваться с выхода блока аналогового ввода на вход блока аналогового вывода (рис. 5.17).

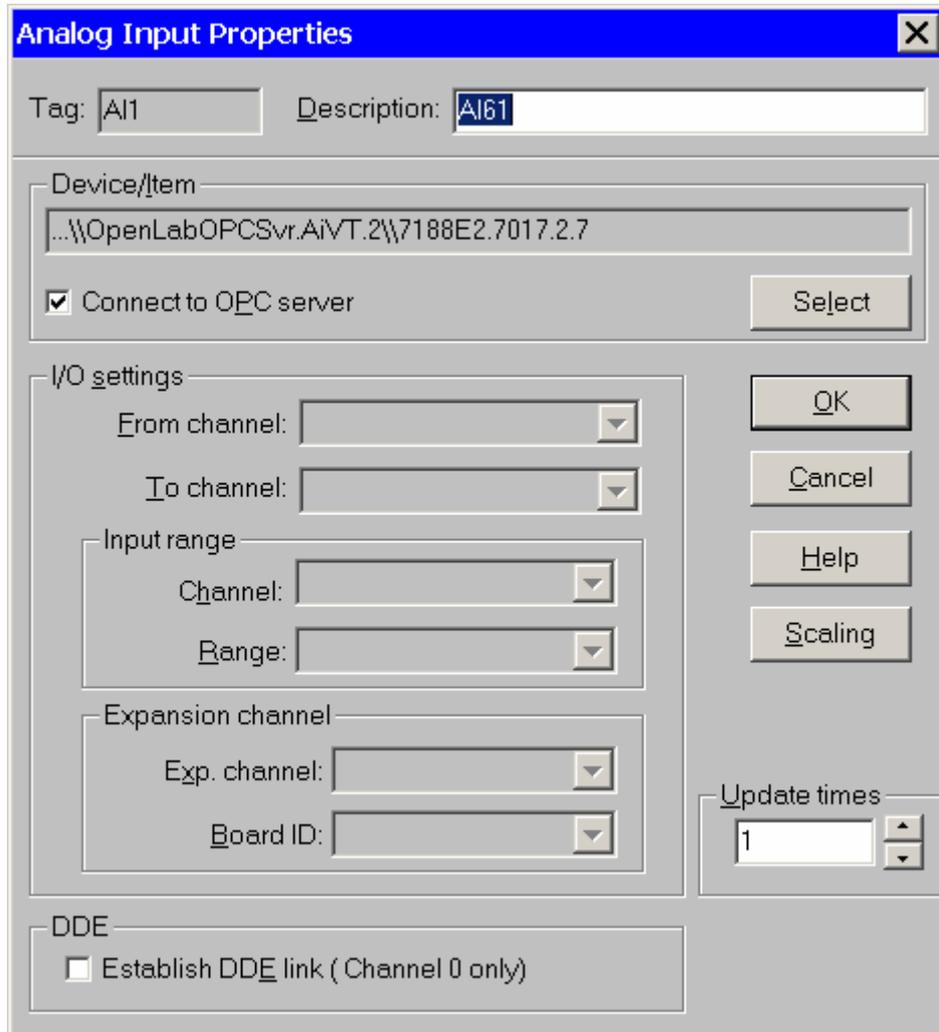


Рис. 5.14. Диалоговое окно конфигурирования блока аналогового ввода

7. *Седьмое действие.* Поместите в окно редактора задач блок дискретного ввода. Для этого на панели инструментов выберите кнопку с мнемоникой **DI** и перетащите ее в окно редактора задач. При этом в окне редактора задач появится блок аналогового вывода с позиционным обозначением **DI**. Произведите двойной щелчок левой кнопкой мыши над этим блоком и аналогичным образом настройте его в соответствии с рис. 5.18. В позиционном обозначении **DI_X0** два последних символа являются обозначением тумблера — источника дискретного сигнала. На приведенном ранее рис. 5.3 приведены обозначения тегов в OPC-сервере, соответствующих имитаторам входных дискретных сигналов.

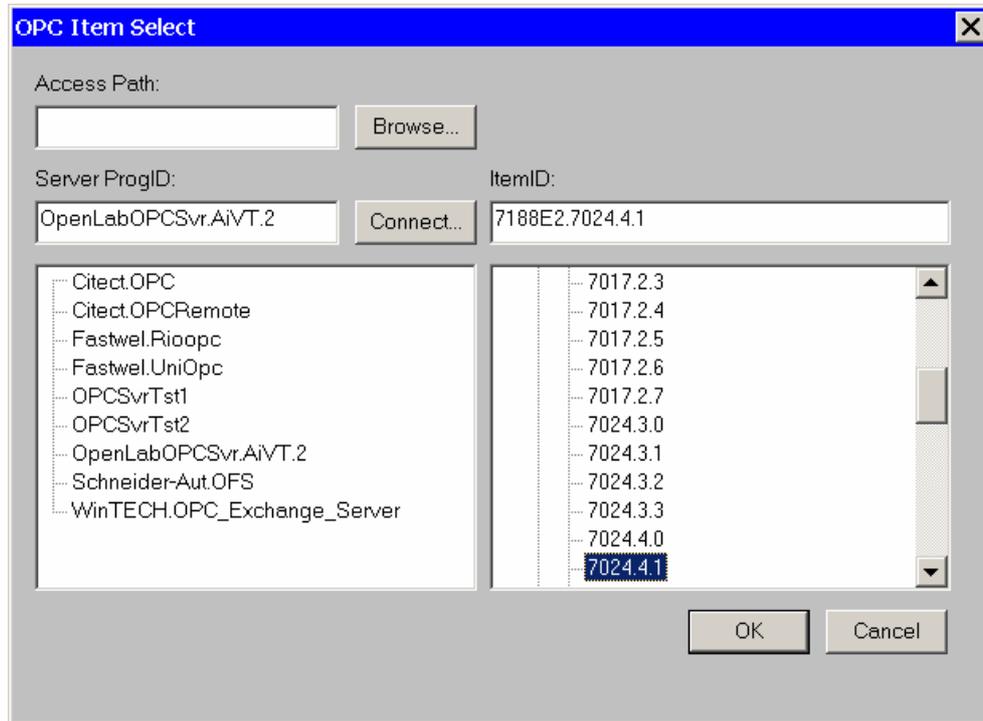


Рис. 5.15. Диалоговое окно выбора параметров OPC-сервера

8. *Восьмое действие.* Поместите в окно редактора задач блок дискретного вывода. Для этого на панели инструментов выберите кнопку с мнемоникой **DO** и перетащите ее в окно редактора задач. При этом в окне редактора задач появится блок аналогового вывода с позиционным обозначением **DO1**. Произведите двойной щелчок левой кнопкой мыши над этим блоком и аналогичным образом настройте его в соответствии с рис. 5.19. В позиционном обозначении **DO_Y0** два последних символа являются обозначением светодиода — приемника дискретного сигнала. На приведенном ранее рис. 5.3 приведены обозначения тегов в OPC-сервере, соответствующих имитаторам выходных дискретных сигналов.
9. *Девятое действие.* Аналогично указанному ранее выполните соединение выхода блока дискретного ввода с входом блока дискретного вывода таким образом, чтобы информация передавалась с блока дискретного ввода на блок дискретного вывода. Таким образом, с помощью указанных блоков дискретного ввода и дискретного вывода будет обеспечена передача информации от тумблера на светодиод.
10. *Десятое действие.* Поместите в окно редактора задач блок дискретного ввода. Для этого на панели инструментов выберите кнопку с мнемоникой **DI** и перетащите ее в окно редактора задач. При этом в окне редактора задач появится

блок аналогового вывода с позиционным обозначением **DI2**. Произведите двойной щелчок левой кнопкой мыши над этим блоком и аналогичным образом настройте его в соответствии с рис. 5.20. В позиционном обозначении **СчетчикDI_X8** два последних символа являются обозначением кнопки — источника дискретного сигнала, для которой ведется подсчет количества нажатий. На приведенном ранее рис. 5.3 приведены обозначения тегов в OPC-сервере, соответствующих имитаторам входных дискретных сигналов.

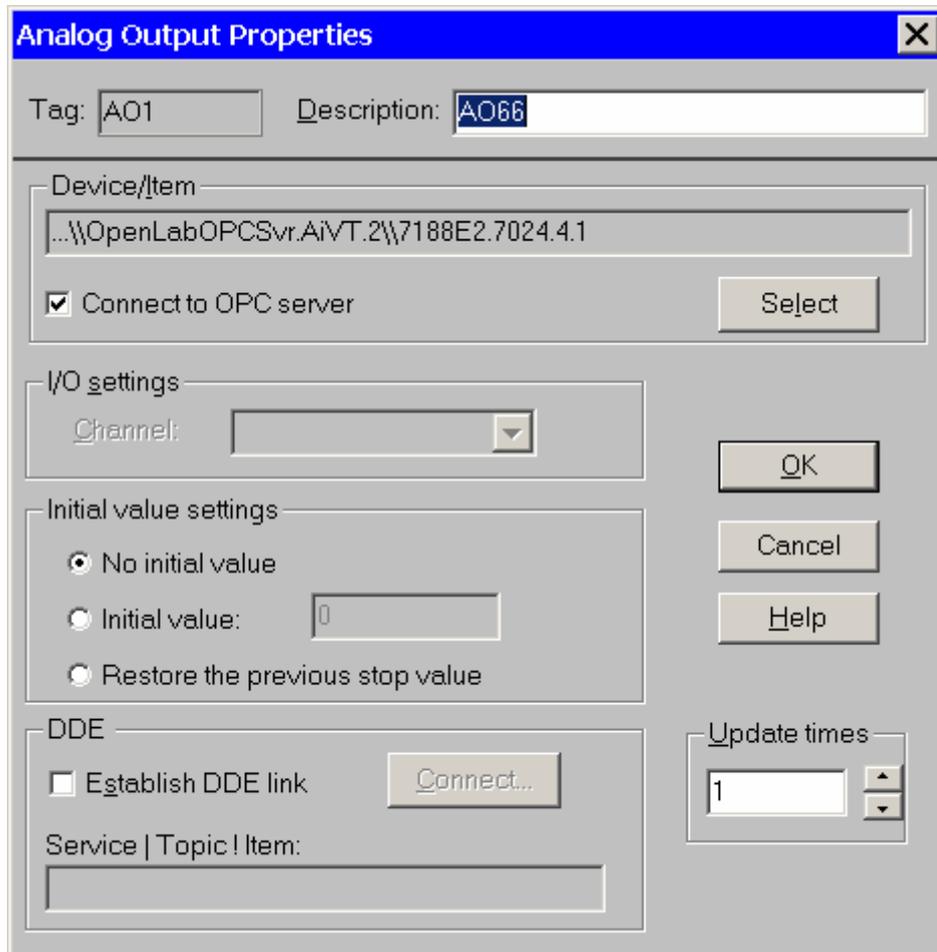


Рис. 5.16. Диалоговое окно конфигурирования блока аналогового вывода

11. *Одиннадцатое действие.* В результате выполненных действий окно редактора задач примет вид, указанный на рис. 5.21.
12. *Двенадцатое действие.* Сделайте активным окно **DISP1** редактора форм отображения. На панели инструментов выберите кнопку **RealTime Trend Graph**

(временной график) и с помощью мыши перетащите ее и положите во внутреннюю область окна редактора форм отображения и, не отпуская левой кнопки мыши, установите нужный размер графика. Для настройки свойств временного графика щелкните дважды левой кнопкой мыши над пиктограммой временного графика. В результате появится диалоговое окно настройки. Для привязки графика к источнику информации в этом окне нажмите кнопку **ADD** и в появившемся окне выполните необходимую настройку для индикации выходного сигнала блока аналогового вывода (рис. 5.22). Затем с помощью кнопки **OK** закройте окно.

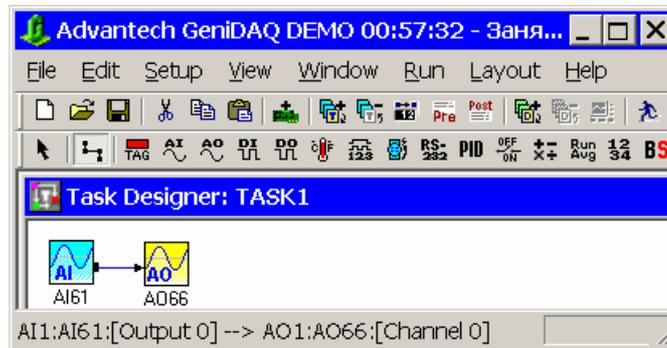


Рис. 5.17. Окно редактора задач

Для отображения содержимого счетчика нажатий кнопки **X8** поместите в окно редактора форм отображения блок цифровой индикации (**Numeric String**) и настройте его в соответствии с рис. 5.23. Снабдите элементы отображения соответствующими надписями (рис. 5.24).

13. *Тринадцатое действие.* Сохраните созданный проект аналогично тому, как это было сделано на занятии 1. После сохранения проекта имеется возможность немедленного запуска созданного проекта с помощью кнопки **Start** на панели инструментов (рис. 5.25). Остановить работу проекта можно с помощью кнопки **Stop** на панели инструментов. В процессе выполнения стратегии рекомендуем выполнить следующие эксперименты. На АВК № 6 подавать на гнездо 1 **Внешние цепи** различные напряжения с выхода источника эталонных напряжений (входной аналоговый сигнал) и измерять напряжение на гнезде 6 (выходной аналоговый сигнал) — эти напряжения должны совпадать. Выполнять переключение тумблера **X0** на имитаторе дискретных сигналов и наблюдать за состоянием светодиода **Y0**. Выполнять нажатие кнопки **X8** на имитаторе дискретных сигналов и наблюдать за состоянием цифрового индикатора.
14. *Пятнадцатое действие.* Завершите работу приложения. Для этого выполните команду **File | Exit**.
15. *Шестнадцатое действие* — выполняет преподаватель. На компьютере, где работает OPC-сервер, командой **File | Exit** завершите его работу и выключите

компьютер. Выключите питание промышленного контроллера. Выключите включенные АВК.

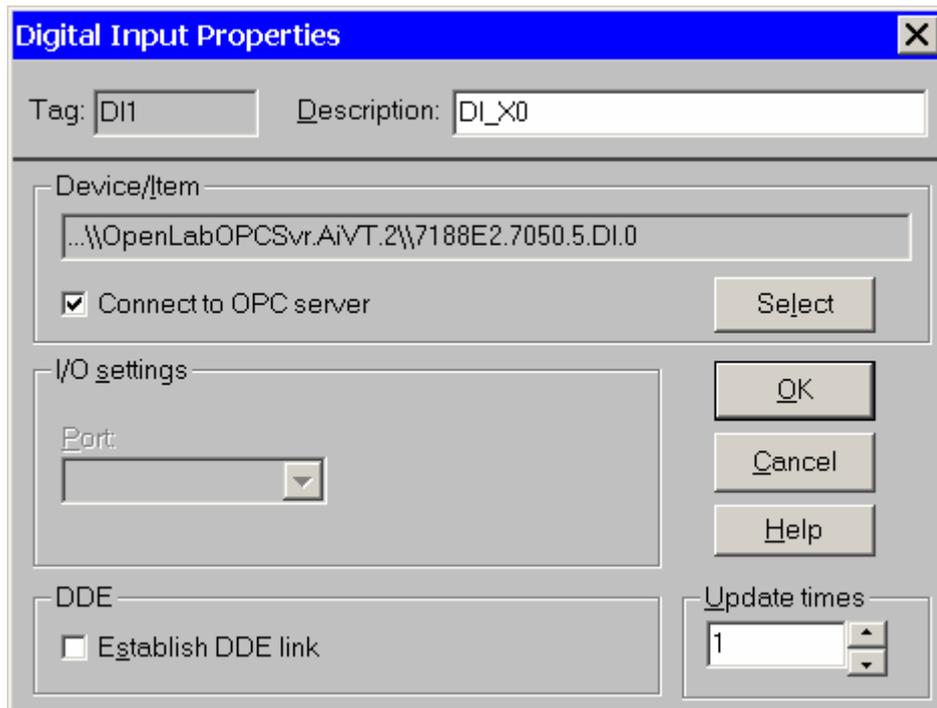


Рис. 5.18. Диалоговое окно конфигурирования блока дискретного ввода

5.4.3. Упражнения

Совет

Обязательно выполните приводимые далее упражнения — это очень важно для практического освоения рассмотренного материала.

Упражнение 4.38. Повтор создания и тестирования приложения "Занятие 12"

Повторите создание и тестирование рассмотренного ранее приложения **Занятие 12**. Для сокращения затрат учебного времени упражнение выполняйте параллельно с преподавателем.

Упражнение 4.39

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование дискретных входов-выходов промышленного контроллера **7188E2** фирмы ICP DAS. В окне

редактора форм отображения снабдить элементы отображения поясняющими надписями.

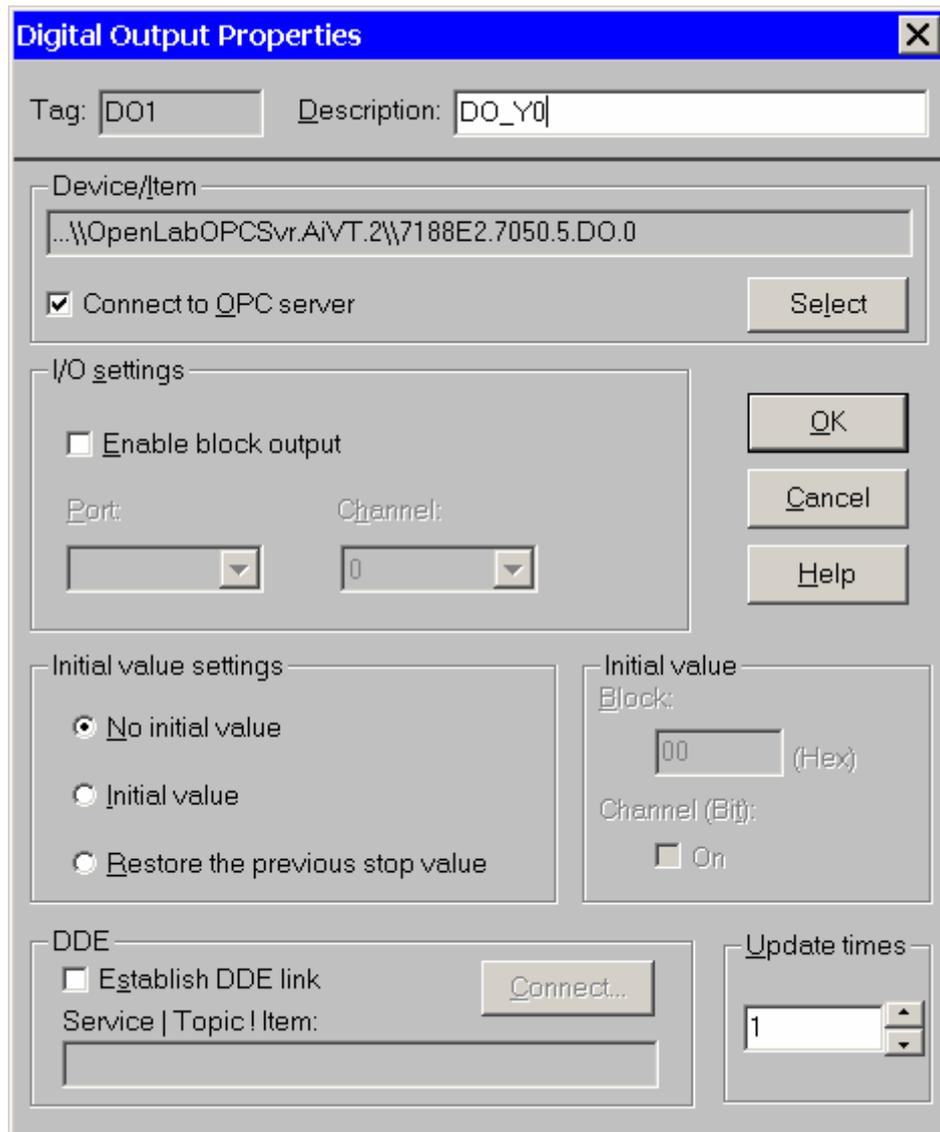


Рис. 5.19. Диалоговое окно конфигурирования блока дискретного вывода

Упражнение 4.40

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование аналоговых

входов-выходов промышленного контроллера 7188E2 фирмы ICP DAS для АВК-1. В окне редактора форм отображения снабдить элементы отображения поясняющими надписями.

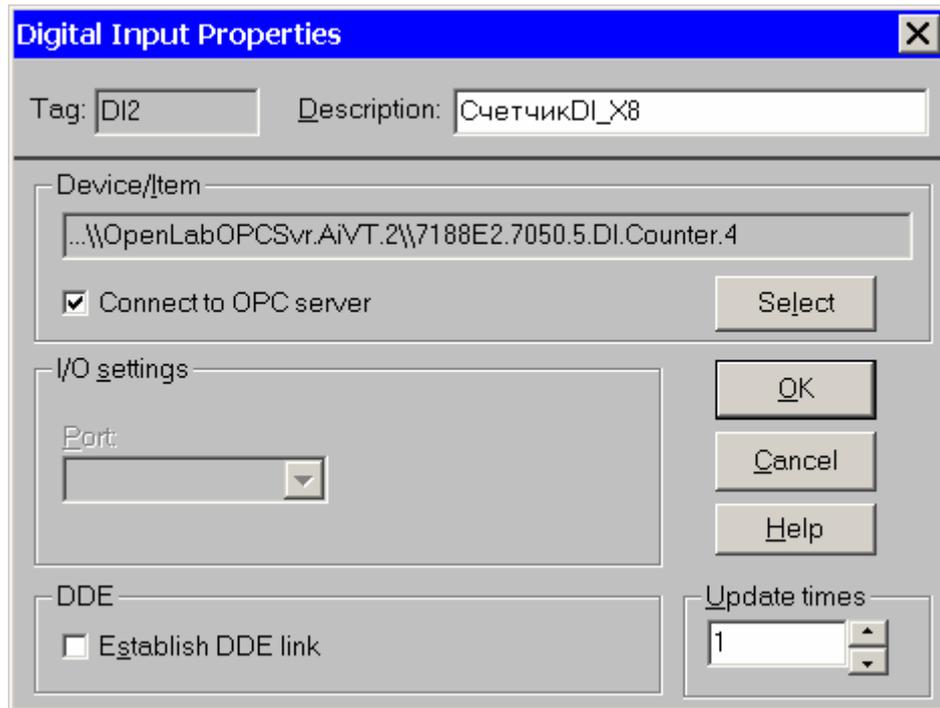


Рис. 5.20. Диалоговое окно конфигурирования блока дискретного ввода

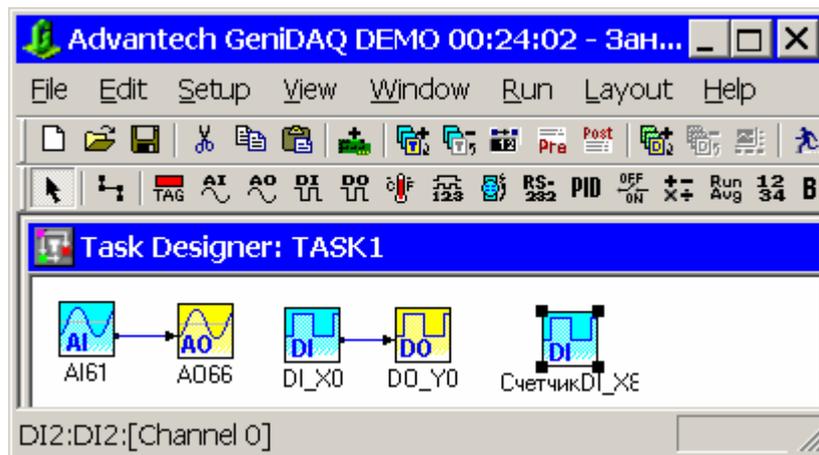


Рис. 5.21. Окно редактора задач

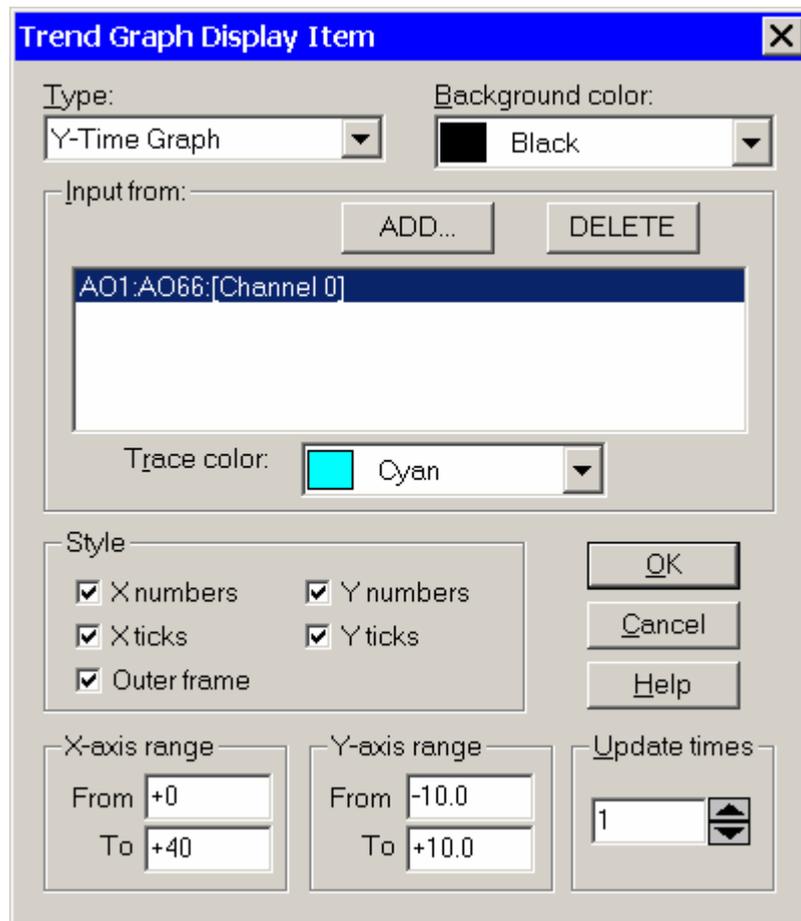


Рис. 5.22. Настройка графика времени

Упражнение 4.41

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование аналоговых входов-выходов промышленного контроллера **7188E2** фирмы ICP DAS для АВК-2. В окне редактора форм отображения снабдить элементы отображения поясняющими надписями.

Упражнение 4.42

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование аналоговых входов-выходов промышленного контроллера **7188E2** фирмы ICP DAS для АВК-3. В

окне редактора форм отображения снабдить элементы отображения поясняющими надписями.

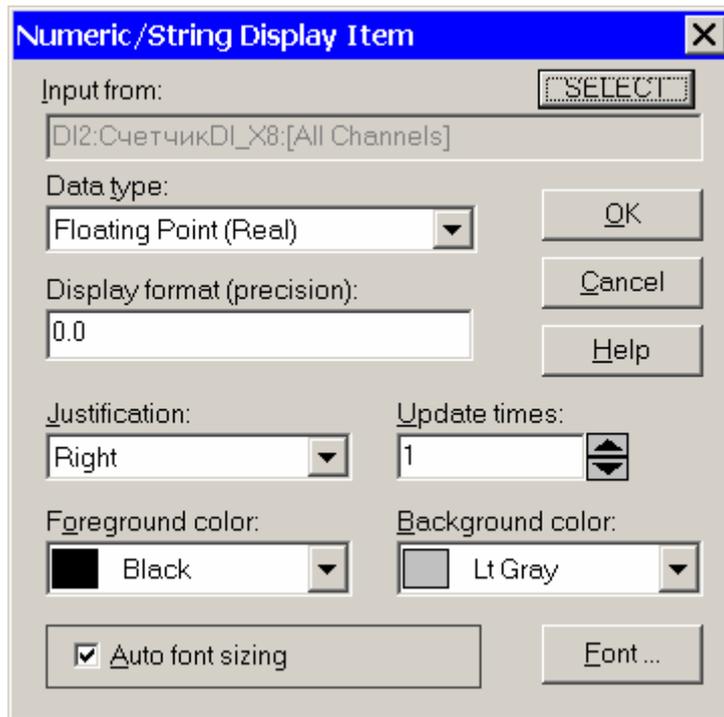


Рис. 5.23. Настройка блока цифровой индикации

Упражнение 4.43

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование аналоговых входов-выходов промышленного контроллера 7188E2 фирмы ICP DAS для АВК-4. В окне редактора форм отображения снабдить элементы отображения поясняющими надписями.

Упражнение 4.44

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование аналоговых входов-выходов промышленного контроллера 7188E2 фирмы ICP DAS для АВК-5. В окне редактора форм отображения снабдить элементы отображения поясняющими надписями.

Упражнение 4.45

Разработать стратегию, обладающую следующими возможностями. Задать период запуска задачи 2 секунды. Стратегия должна обеспечивать тестирование аналоговых входов-выходов промышленного контроллера 7188E2 фирмы ICP DAS для АВК-6. В окне редактора форм отображения снабдить элементы отображения поясняющими надписями.

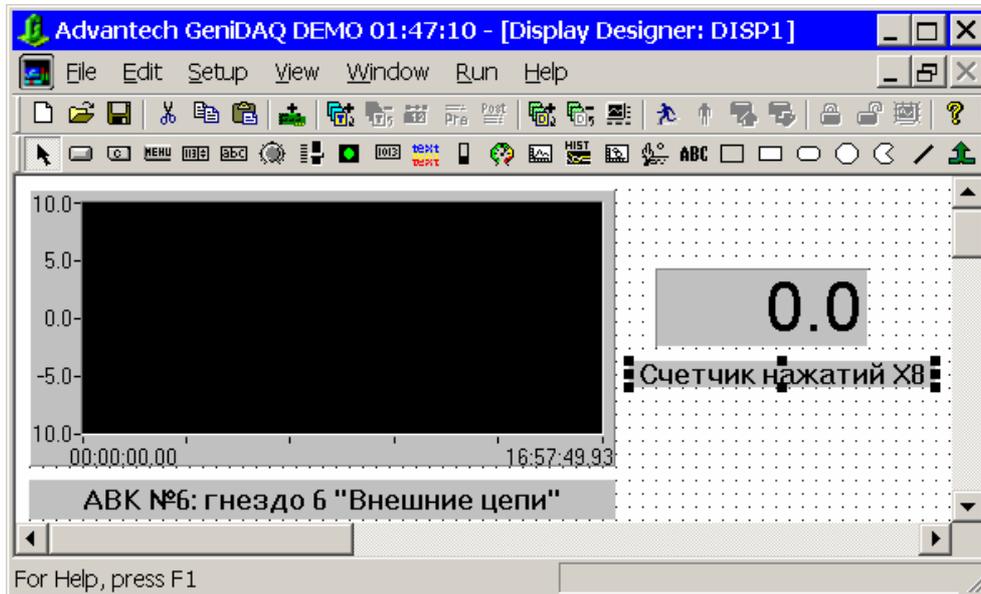


Рис. 5.24. Вид окна редактора форм отображения

Совет

Для лучшего освоения изучаемого материала советуем вам дополнить указанные ранее материалы курсовым проектированием. Варианты заданий для курсового проектирования приводятся в приложении 1.

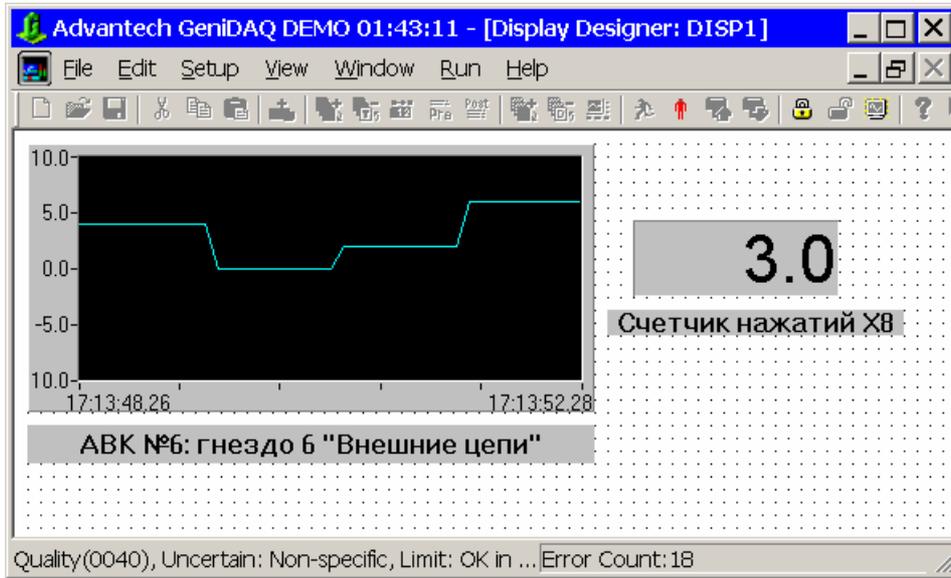


Рис. 5.25. Вид окна редактора форм отображения

Литература

1. Чьонг Д.Т. Взаимодействие открытых систем промышленной автоматизации — состояние и проблемы // Информационно-управляющие системы, 2003. №2-3.
2. Системы диспетчерского управления и сбора данных (SCADA-системы) // Мир компьютерной автоматизации, 1999. №1.
3. Куцевич Н.А. SCADA-системы. Взгляд со стороны // Промышленные контроллеры и АСУ, 1999. №1.
4. Куцевич Н.А. SCADA-системы, или муки выбора // Мир компьютерной автоматизации, 1999. №1.
5. Genie 3.0. Руководство пользователя, Prosoft Ltd., 1998.
6. Куцевич Н.А. Citect - новая SCADA-система на российском рынке и новые возможности // Промышленные контроллеры и АСУ, 2000. №1.
7. Калядин А.Ю. Scada-система Citect — что внутри? // PCWeek, 1999. №48.
8. SCADA-продукты на российском рынке // Мир компьютерной автоматизации, 1999. №1.
9. Давыдов В.Г., Чьонг Д.Т. OPC-серверы с открытой архитектурой — средства взаимодействия компонентов в промышленной автоматизации // Автоматизация в промышленности, 2003. №7.
10. Теркель Д. OLE for Process Control — свобода выбора // Современные технологии автоматизации, 1999. №3.
11. Кузнецов А. SCADA-системы: программистом можешь ты не быть // Современные технологии автоматизации, 1996. №1.
12. Advantech GeniDAQ User's Manual. Advantech Co., 2000.
13. Давыдов В.Г., Чьонг Д.Т. Проектирование компьютерных систем управления на основе SCADA-систем: Учеб. пособие. СПб.: Изд-во СПбГПУ, 2004. 152 с. (современные компьютеризованные образовательные технологии). ISBN 5-7422-0573-2.
14. Гарнаев А.Ю. Самоучитель VBA. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2004. — Ил.
15. Карпов Б. VBA: специальный справочник. — СПб: Питер, 2002. — 416 с.: ил.

МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «Уральский государственный горный университет»



УТВЕРЖДАЮ
Проректор по учебно-методической
работе В. В. Зубов

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Б1.В.09.13 ПРОМЫШЛЕННЫЕ МЕХАТРОННЫЕ СИСТЕМЫ

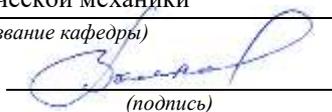
*Специальность –
21.05.04 Горное дело*

*Специализация –
Мехатроника и робототехника промышленных комплексов*

Одобрена на заседании кафедры

Технической механики
(название кафедры)

Зав. кафедрой


(подпись)

Волков Е. Б.
(Фамилия И.О.)

Протокол № 1 от 13.09.2024
(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического
(название факультета)

Председатель


(подпись)

Осипов П.А.
(Фамилия И.О.)

Протокол № 2 от 18.10.2024
(Дата)

Екатеринбург

Лабораторная работа №1

Изучение режимов работы робота–штабелера

Цель работы:

Ознакомиться с типами управления и изучить режимы работы робота–штабелера используя «сокращенный» и «расширенный» функционал управления роботом.

Теоретическая часть:

Интерфейс управления роботом–штабелером может отображаться в двух различных видах: «сокращенный» и «расширенный». Вне зависимости от режима отображения робот–штабелер может управляться в тремя способами:

1. Ручное управление, которое осуществляется путем перемещения оператором «ползунков», задающих значения координат, и нажатия кнопки перемещений (рисунок 1.1).



Рисунок 1.1 – Интерфейс ручного управления роботом–штабелером

2. Управление при помощи командной строки, в которой оператор прописывает текст команды, выполняемой роботом после нажатия клавиши «Enter» (рисунок 1.2).



Рисунок 1.2 – Интерфейс управления при помощи командной строки

3. Управление роботом–штабелером при помощи управляющей программы, созданной в текстовом файле и загруженной оператором.

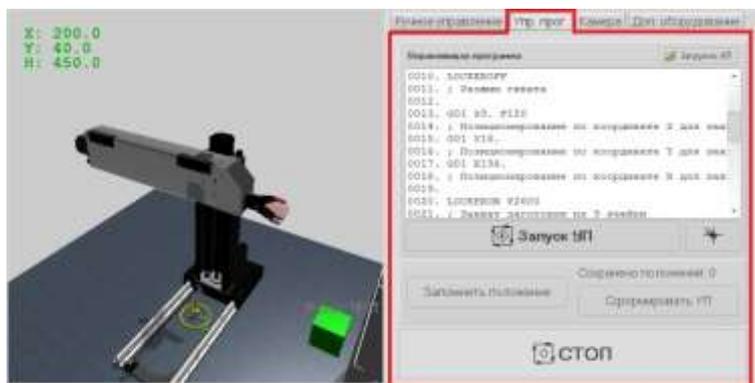


Рисунок 1.3 – Интерфейс управления роботом–штабелером при помощи управляющей программы

На рисунках 1.1 – 1.3 представлен «сокращенный» вид управления роботом–штабелером. В «расширенном» виде в интерфейсе управления роботом–штабелером отсутствует графическое представление положения робота в пространстве. Интерфейс управления роботом в «расширенном» виде представлен на рисунке 1.4.

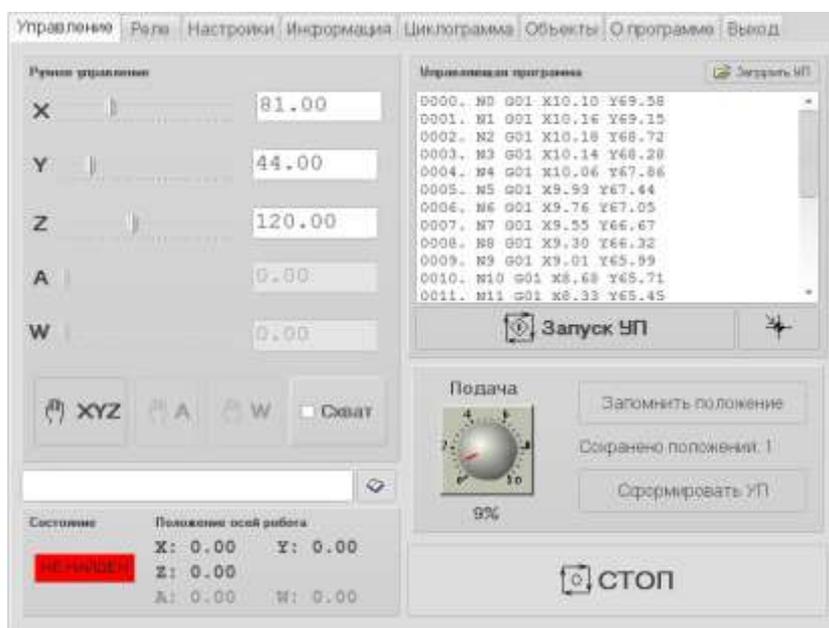


Рисунок 1.4 – Интерфейс управления роботом в «расширенном» виде

Для ручного управления роботом оператор должен переместить «ползунок» выбранной координаты (или нескольких координат для одновременного перемещения) в положение, соответствующее заданному положению робота. Значение координаты можно увидеть в белом поле рядом с выбранным «ползунком».

Для того, чтобы робот переместился в заданные координаты, необходимо после их назначения нажать кнопку XYZ. Для выхода звеньев робота в нулевое положение необходимо нажать на кнопку . Далее выбрать из списка, по какой координате следует произвести выход в нулевое положение. Для управления схватом необходимо поставить либо убрать галочку в поле рядом со словом СХВАТ.

Для управления роботом–манипулятором при помощи командной строки необходимо написать текст команды в поле командной строки (см. рисунок 1.2).

Основной командой для управления звеньями робота является команда линейного перемещения G01. Пример текста команды выглядит следующим образом:

G01 X50. Y100. Z30. F100,

где G01 – команда линейного перемещения в заданные координаты;

X50. Y100. Z30. – координаты конечных положений звеньев робота;

F100 – обозначение скорости перемещения, цифры после буквы указывают на процент от максимального перемещения.

Перемещение можно произвести по одной, двум или трем координатам одновременно. Координаты X, Y и Z задаются в миллиметрах.

Для управления схватом используются команды LOCKERON и LOCKEROFF.

Команда LOCKERON сжимает губки схвата, команда LOCKEROFF – разжимает.

Для выхода звеньев робота в нулевое положение используется команда RHOME. При этом необходимо указать координату, для которой указывается команда, например:

RHOMEX – выход в нулевое положение по оси X;

RHOMEY – выход в нулевое положение по оси Y;

RHOMEZ – выход в нулевое положение по оси Z.

Все команды управления роботом-манипулятором записываются ПРОПИСНЫМИ буквами.

Более подробно о программировании можно узнать из прилагаемого пособия по программированию.

Для управления роботом–штабелером при помощи управляющей программы (УП), ее нужно загрузить нажатием на кнопку .

Перед загрузкой УП необходимо предварительно составить текст программы в текстовом редакторе «Блокнот», в формате «имя программы.txt». Каждая команда с параметрами перемещения записывается с новой строки. Текст после знака « ; » не считывается и является комментарием.

Также составление УП возможно перемещением робота в нужные положения и запоминанием полученных координат с помощью кнопки «Запомнить положение».

После завершения необходимых перемещений робота нужно нажать кнопку «Сформировать УП», и управляющая программа будет создана автоматически. При этом нужно указать путь сохранения и дать ей имя. Эту программу можно открыть текстовым редактором и отредактировать.

Ход работы

- 1) Включить блоки управления стендом.
- 2) Нарисовать схему подключения элементов стенда.
- 3) Запустить программу управления штабелером. В открывшемся окне проверить, что строка «состояние» горит зеленым цветом и есть надпись «Найден».
- 4) При помощи вкладки «объекты» в расширенном интерфейсе управления роботом создать минимум пять объектов, имитирующих ячейки для заготовок на складе.
- 5) В сокращенном режиме работы, оставаясь в режиме имитатора, произвести перемещения робота таким образом, чтобы схват робота поочередно оказывался там, где расположены созданные объекты. Перемещения необходимо осуществлять по одной оси, двум осям и трем осям одновременно, задавая при этом различные скорости перемещения звеньев робота–манипулятора по осям. Последовательность перемещения необходимо записать в таблицу 1.

Таблица 1 – Последовательность перемещения робота–манипулятора

№ перемещения	Координата X	Координата Y	Координата Z	Скорость перемещения F
1				
2				
3				
4				
5				

Вывести робот–манипулятор в нулевое положение по всем осям.

Составить для каждого перемещения команду. Записать текст команды в таблицу 2.

Таблица 2 – Команды перемещения робота–манипулятора

№ перемещения	Текст команды
1	
2	
3	
4	
5	

6) Путем последовательного ввода команд в командную строку произвести перемещения звеньев робота по осям. Обращать внимание на правильность написания команд. Отслеживать текущие координаты осей робота по значениям в поле «Положение осей робота». После окончания каждого перемещения нужно запоминать положение робота–манипулятора в программе нажатием кнопки «Запомнить положение» на вкладке «Управляющая программа». После запоминания всех перемещений по этим опорным точкам сформировать управляющую программу нажатием кнопки «Сформировать УП».

7) Открыть сформированную УП, добавить в конце программы команды выхода в нулевое положение по всем осям. Сохранить файл и запустить управляющую программу в имитаторе.

8) После правильного выполнения этой программы открыть её текст. Для имитации переноса заготовки из одной позиции в другую, перед перемещением робота в первую позицию вставить команду на разжим схвата, при достижении роботом первой позиции вставить команду на зажим схвата, имитируя зажим детали. Затем после перемещения робота во вторую позицию вставить команду на разжим схвата, имитируя укладку детали в ячейку.

9) По той же схеме произвести имитацию перемещения заготовки по трем ячейкам, при этом во второй ячейке происходит сначала разжим схвата и укладывание заготовки в ячейку, затем снова следует зажим этой заготовки и перенос в последнюю ячейку.

9) Записать управляющие программы. Составить отчет о выполненной работе.

Лабораторная работа №2

Изучение интерпретатора языка программирования робота

Цель работы:

Изучить понятие интерпретатора. Понять процесс работы интерпретатора. Определить работу интерпретатора при написании управляющей программы робота.

Теоретическая часть:

Программирование систем с числовым программным управлением (ЧПУ) производится посредством G – кода. G – код – это общее название языка программирования систем с ЧПУ, так как он имеет множество реализаций и дополнений у разных производителей и для разных устройств. Программа, написанная с использованием G-кода, состоит из кадров, каждый кадр содержит набор команд управления. Интерпретатор анализирует и тут же выполняет (собственно интерпретация) программу покомандно (или построчно), по мере поступления её исходного кода на вход интерпретатора.

Алгоритм работы простого интерпретатора включает в себя:

- 1) Прочитать инструкцию.
- 2) Проанализировать инструкцию и определить соответствующие действия.
- 3) Выполнить соответствующие действия.
- 4) Если не достигнуто условие завершения программы, прочитать следующую инструкцию и перейти к пункту 2.

По сути робот–манипулятор – это перепрограммируемый автоматический манипулятор промышленного применения. Под перепрограммируемостью, в соответствии со стандартом, понимается свойство промышленного робота заменять управляющую программу автоматически или при помощи человека–оператора.

К перепрограммированию относится изменение последовательности и(или) величин перемещений по степени подвижности и управляющих функций с помощью средств управления на пульте устройства управления.

Работа интерпретатора заключается в сопоставлении команд G – кода с командами, подаваемыми на микроконтроллер драйвера управления звеньями робота–манипулятора. При этом, если команды управления записаны неверно, интерпретатор не может сопоставить их с командами для микроконтроллера и действия не происходит.

Интерпретатор языка программирования встроен в программу для управления роботом, поэтому правильность работы интерпретатора определяется правильностью исполнения команд управляющей программы роботом–манипулятором.

Ход работы:

- 1) Включить блоки управления стендом, запустить программу управления стендом;
- 2) Выйти из режима «имитатора», убрав галочку в поле рядом с соответствующей надписью;
- 3) Вывести оси робота в нулевые положения;
- 4) Используя данные таблицы 2 из лабораторной работы №1 выполнить команды на скорости перемещения 20%, путем последовательного ввода команд в командную строку произвести перемещения звеньев робота–манипулятора по осям. Обращать внимание на правильность написания команд. Отслеживать текущие координаты осей робота по значениям в поле «Положение осей робота». После окончания каждого перемещения нужно запоминать положение робота–манипулятора в программе нажатием кнопки «Запомнить положение» на вкладке «Управляющая программа». После запоминания всех перемещений по этим опорным точкам сформировать управляющую программу нажатием кнопки «Сформировать УП». При этом обращать внимание, чтобы звенья робота при перемещении не наезжали на элементы стенда;
- 5) Открыть сформированную УП, добавить в конце программы команды выхода в нулевое положение по всем осям. Сохранить файл и запустить управляющую программу;
- 6) Описать действия робота–манипулятора для выполнения данной управляющей программы используя алгоритм работы простого интерпретатора;
- 7) Составить отчет о выполненной работе.

Лабораторная работа №3.

Изучение команд перемещения робота–манипулятора. Изучение работы робота–штабелера в декартовой системе координат

Цель работы

Изучение команд перемещения звеньев робота в декартовой системе координат.

Теоретическая часть

Перемещение звеньев робота задается тремя составляющими:

- координата X робота, передвижение горизонтального звена робота со схватом вперед–назад;
- координата Y робота, изменение высоты горизонтального звена робота вдоль вертикального звена вверх–вниз;
- координата Z робота, передвижение робота вправо–влево.

Данные координаты имеют только положительные значения, отсчитываемые от датчиков нулевого положения. На рисунке 3.1 звенья робота располагаются в нулевых координатах.

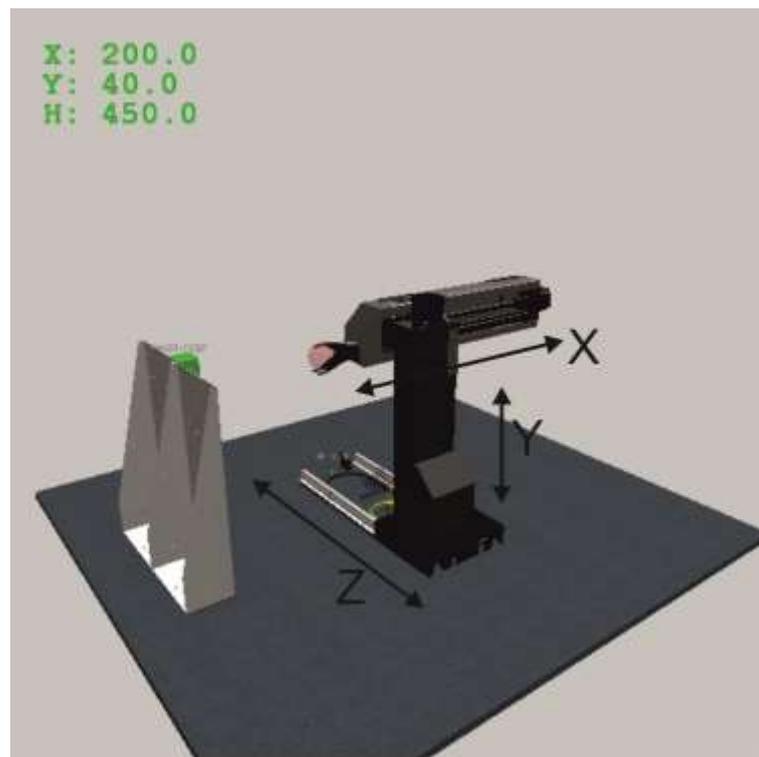


Рисунок 3.1 – Координаты перемещений звеньев робота

Декартова система координат показана на рисунке 3.2. В этой системе координат определяется положение центра схвата робота–штабелера. Декартовы координаты имеют только положительные значения по осям X и Z . По оси Y координаты имеют положительные и отрицательные значения. Положение центра схвата в декартовой системе отображается координатами в верхнем левом углу графического окна.

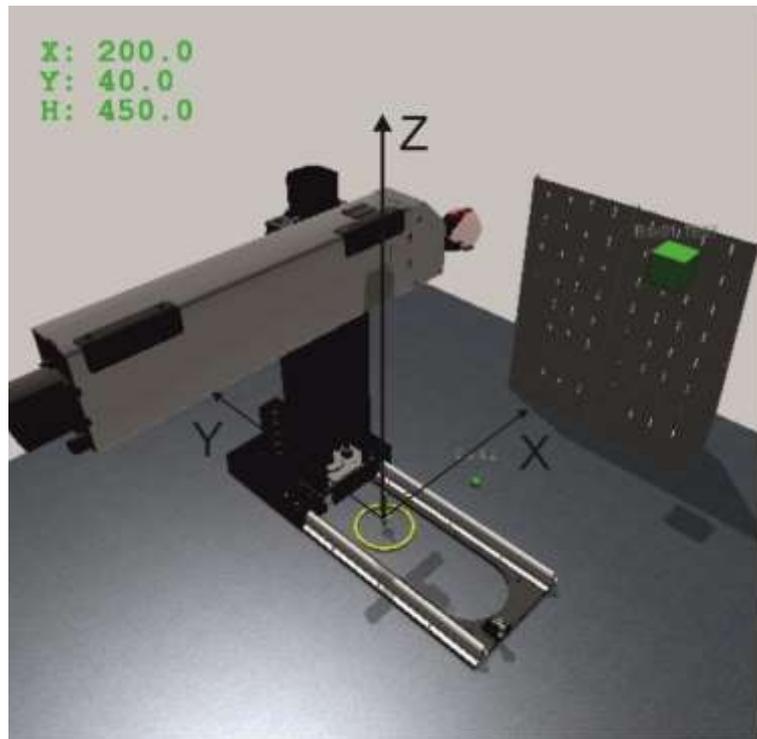


Рисунок 3.2 – Декартова система координат робота – штабелера

Более подробно с командами, используемыми в стенде можно ознакомиться в пособии по программированию.

Ход работы:

- 1) Включить блок управления стендом.
- 2) Запустить программу управления. В открывшемся окне проверить, что строка «состояние» горит зеленым цветом и есть надпись: «Найден».
- 3) Задавая различные величины перемещений звеньев робота определить зону работы робота в декартовых координатах и крайние точки его перемещения. Записать их в значениях (X, Y, Z) и для каждой точки написать соответствие в собственных перемещениях звеньев. Оформить в виде таблицы.
- 4) Установить кронштейны в произвольные ячейки склада. Зажать в схвате робота заготовку. Переместить поочередно схват робота в ячейки. Записать декартовы координаты ячеек. В графической части создать объекты, имитирующие зону расположения установленных ячеек. Записать полученные координаты объектов.
- 5) Определить, совпадают ли координаты перемещений робота и расположения ячеек.
- 6) Составить отчет о выполненной работе.

Лабораторная работа №4

Синтез программы сортировки объектов

Цель работы

Управление роботом–штабелером при создании управляющей программы сортировки объектов.

Теоретическая часть:

Для сортировки объектов по стенду представлена программа:

; По программе робот берет заготовку диаметром 35 миллиметров из 9 ячейки и ставит заготовку в 16 ячейку. Заготовки на длинном кронштейне.

RHOMEX

; Выход в нулевое положение по координате X

RHOMEY

; Выход в нулевое положение по координате Y

RHOMEZ

; Выход в нулевое положение по координате Z

LOCKEROFF

; Разжим схвата

G01 Z9. F150

; Позиционирование по координате Z для захвата заготовки из 9 ячейки

G01 Y18.

; Позиционирование по координате Y для захвата заготовки из 9 ячейки

G01 X196.

; Позиционирование по координате X для захвата заготовки из 9 ячейки

LOCKERON P2600

; Захват заготовки из 9 ячейки

G01 Y5.

; Движение вверх по координате Y для безопасного выхода из позиции захвата заготовки

G01 Z275.

; Позиционирование по координате Z для установки заготовки в 16 ячейку

G01 Y70.

; Позиционирование по координате Y для установки заготовки в 16 ячейку

LOCKEROFF P2000

; Разжим схвата на 2000 импульсов

G01 X150. F150

; Безопасный отход по координате X из 16 ячейки

RHOMEX

; Выход в нулевое положение по координате X

RHOMEY

; Выход в нулевое положение по координате Y

RHOMEZ

; Выход в нулевое положение по координате Z

Ход работы

1) Включить блок управления стандом;

2) Запустить программу управления стандом. В открывшемся окне проверить, что строка «состояние» горит зеленым цветом и есть надпись: «Найден». Установить кронштейны в ячейки 9 и 16.

3) Разобрать представленную выше управляющую программу по сортировке объектов.

4) Создать текстовый документ формата «имя_программы.txt». Записать в документ строки представленной управляющей программы, которые содержат исполнительные команды. Сохранить текстовый документ.

5) Выйти из режима «имитатор», переместить робот–штабелер в ячейки 9 и 16, при этом проверить, чтобы схват робота располагался точно в ячейках. Сравнить полученные координаты с координатами в управляющей программе. При необходимости ввести коррективы в программу.

6) В командную строку поочередно в порядке расположения в тексте вставлять строки с командами управляющей программы, выполняя каждую команду отдельно. Визуально следить за правильностью выполнения команды. Не допускать аварийных ситуаций.

7) Импортировать отработанную управляющую программу в программу управления стандом.

8) Запустить управляющую программу, контролируя визуально перемещения робота. Не допускать аварийных ситуаций.

9) После успешного завершения программы составить аналогичную для перемещения заготовки по трем ячейкам.

10) Составить отчет о выполненной работе.

Лабораторная работа №5

Синтез программы оптимизированного складирования объектов

Цель работы: произвести оптимизацию управляющей программы из лабораторной работы №4.

Теоретическая часть:

Оптимизация процесса складирования объектов подразумевает сокращение времени на выполнение автоматизированной системой своей управляющей программы. Одним из способов, позволяющих оптимизировать производственный процесс, является моделирование (имитация) выполнения программы. Данный вид оптимизации можно применить, используя перед выполнением управляющей программы среду «имитатор» для отработки управляющей программы. После оптимизации в «имитаторе» отработанную программу можно запустить для исполнения автоматизированной системой.

Для оптимизации также имеет место замена поочередного перемещения по нескольким осям одним совместным перемещением осей (интерполяция). Например, использовать одну команду G01 X46. Y120. F100 вместо двух последовательных команд G01 X46. F100 и G01 Y120. F100.

Еще одним примером оптимизации является использование меток для перехода, если нужно сделать программу повторяющейся, то есть запустить цикл.

Пример программы:

RHOME X

RHOME Y

RHOMR Z

1000:

G01 X100. Y100. F100

G01 Z150. F150

G01 X50. Y150. F100

G01 Z250. F150

RHOME X

RHOME Y

RHOMR Z

SHIFТТО P1000

В данной программе использована метка перехода 1000 в четвертой строке и обращение к ней в конце программы командой SHIFТТО P1000 в конце программы. То есть программа первый раз пройдет все строки и выполнит переход к четвертой

строке и в дальнейшем не будет выполнять первые три строчки. После числового обозначения метки ставится знак «:».

Ход работы:

- 1) Включить блок управления стендом.
- 2) Запустить программу управления штабелером. В открывшемся окне проверить, что строка «состояние» горит зеленым цветом и есть надпись «Найден».
- 3) В режиме «имитатор» запустить управляющую программу, полученную в лабораторной работе №4. Произвести оптимизацию данной программы.
- 4) Выполнить зацикливание данной программы.
- 5) Запустить получившуюся программу складирования объектов. При этом следует подавать заготовки в первоначальную ячейку.
- 6) Составить отчет о выполненной работе.

ТВЕРЖДАЮ
Проректор по учебно-методической
работе В. В. Зубов



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Б1.В.09.15 ИНФОРМАЦИОННАЯ ПОДДЕРЖКА МЕХАТРОННЫХ ПРОИЗВОДСТВ

Специальность –

21.05.04 Горное дело

Специализация –

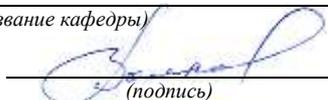
Мехатроника и робототехника промышленных комплексов

Одобрена на заседании кафедры

Технической механики

(название кафедры)

Зав. кафедрой


(подпись)

Волков Е.Б.

(Фамилия И.О.)

Протокол № 1 от 13.09.2024

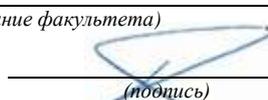
(Дата)

Рассмотрена методической комиссией
факультета

Горно-механического

(название факультета)

Председатель


(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Лекция 1. Общие понятия об информационной поддержке мехатронных производств	7
1.1. Этапы жизненного цикла промышленных изделий	9
1.2. Системный подход к информационной поддержке сложных систем: анализ, синтез, моделирование, оптимизация [2, 3]	14
Контрольные вопросы	27
Лекция 2. Машиностроительные и металлообрабатывающие производства	29
2.1. Гибкие автоматизированные системы и производства	30
2.2. Автоматизированные линии	38
2.3. Гибкие производственные системы механообрабатывающего производства	50
Контрольные вопросы	53
Лекция 3. Введение в процесс проектирования мехатронных систем	55
3.1. Основные принципы проектирования	56
3.2. Схема процесса проектирования	60
3.3. Системный подход к проектированию	67
Контрольные вопросы	76
Лекция 4. Системы автоматизированного проектирования в машиностроении	78
4.1. Общая характеристика систем автоматизированного проектирования	78
4.2. Структура и разновидности систем автоматизированного проектирования	113
4.3. Программные системы проектирования (примеры программ) [3]	116
Контрольные вопросы	124
Лекция 5. Интегрированный подход к разработке мехатронных систем	125
5.1. Системы PDM	126

5.2. Метод обмена данными	129
5.3. Тенденции в развитии интегрированных систем [10, 11] ...	132
Контрольные вопросы	133

Лекция 6. Концептуальное проектирование и модернизация мехатронных систем	135
6.1. Основные положения	135
6.2. Инструментальные средства концептуального проектирования и модернизации систем	139
Контрольные вопросы	148

Лекция 7. Информационная поддержка жизненного цикла мехатронных систем	150
7.1. CALS-технологии (основные понятия) [1, 3]	150
7.2. STEP-стандарты [5]	158
7.3. Организация в STEP информационных обменов [1]	160
7.4. Проблемы практического использования CALS-технологий	162
Контрольные вопросы	165

Лекция 8. Основные прикладные методы и средства построения интегрированных мехатронных систем	167
8.1. Формирование общих проектных решений [3, 12, 13]	167
8.2. Построение модели процесса разработки мехатронных систем	170
Контрольные вопросы	188

Лекция 9. Автоматизированные системы управления мехатронными производствами [1, 3, 18]	190
9.1. Автоматизация управления предприятиями (АСУП)	190
9.2. Автоматизация управления технологическими процессами (АСУТП)	192
9.3. Информационные логистические системы	194
9.4. Автоматизированные системы делопроизводства	197
Контрольные вопросы	198

Заключение	200
------------------	-----

Библиографический список	201
--------------------------------	-----

ВВЕДЕНИЕ

Информационная поддержка процессов создания и эксплуатации мехатронных (сложных) производств базируется на широком классе универсальных автоматизированных систем, предназначенных для использования на всех этапах жизненного цикла изделий (ЖЦИ). На этапах проектирования и подготовки производства используются интегрированные САПР – CAD, CAM, CAE, PDM, CAPP и др.; на этапа производства и реализации изделий – SCADA, CNC, MRP, ERP, MES и др.; на этапе эксплуатации и утилизации изделий – CRM, SQSM, MES.

Мировой опыт промышленного производства сложных изделий на современном уровне свидетельствует, что их создание невозможно без использования информационных систем поддержки производственных процессов. Такая поддержка получила название CALS (русская аббревиатура КСПИ) – Continuous Acquisition and Lifecycle Support (компьютерное сопровождение и поддержка жизненного цикла изделий). Данная система поддержки производственных процессов призвана заменить традиционную технологию создания и эксплуатации производственных процессов, которая отличается фрагментарным использованием автоматизированных систем. Она допускает дублирование и несогласованность данных, возможность рассмотрения недостаточно перспективных вариантов производственных процессов, в то время как лучший вариант остается нерассмотренным. При этом возможно недостаточно эффективное использование знаний, накопленных в процессе создания и эксплуатации производственных процессов [19].

Основное назначение CALS-технологий – обеспечение единого информационного пространства (ЕИП) и технологий для предоставления необходимой информации в нужное время, в нужном виде, в конкретном месте любому из участников жизненного цикла (ЖЦ) производственного процесса.

CALS-технологии развиваются прежде всего на базе САПР (на стадии создания мехатронных систем), хотя при этом рассматриваются также вопросы моделирования приложений: на начальных этапах моделирования сложных слабоструктурированных приложений используются методы функционального (IDEF0) и информационного (IDEF1X) моделирования систем и методы объективного моделирования на базе среды UML.

Информационная поддержка производственных процессов на стадии их эксплуатации осуществляется за счет применения автоматизированных систем управления мехатронными производствами (АСУП, АСУТП, логистических систем, автоматизированных систем делопроизводства).

Для успешного применения современных информационных технологий в промышленности и других отраслях народного хозяйства необходимо иметь квалифицированные кадры, знающие и умеющие использовать информационные технологии создания и эксплуатации сложных систем различного назначения.

Данный курс лекций направлен на базовую подготовку таких специалистов.

Курс лекций «Информационная поддержка мехатронных комплексов и производств» предназначен в первую очередь для бакалавров и магистров по направлению подготовки 15.03.06 – Мехатроника и робототехника, а также других технических направлений и специальностей, связанных с созданием и управлением сложными техническими системами.

В основу данных лекций положен опыт преподавания информационных технологий (отраженных в учебниках, монографиях и публикациях в научно-технических журналах) в ряде ведущих университетов России – МГТУ им. Н.Э. Баумана, МГТУ «Станкин», УГАТУ (г. Уфа), МИРЭА и других.

Курс содержит 9 лекций, в которых отражены отдельные аспекты информационной поддержки сложных мехатронных систем и производств.

Авторы выражают благодарность рецензентам и всем коллегам за советы и рекомендации при обсуждении материала лекций.

Лекция 1

ОБЩИЕ ПОНЯТИЯ ОБ ИНФОРМАЦИОННОЙ ПОДДЕРЖКЕ МЕХАТРОННЫХ ПРОИЗВОДСТВ

Современные производства сложной техники являются базой инновационной экономики, основанной для создания и производства наукоемких изделий, конкурентоспособных на мировых рынках.

Сложные производства базируются на согласованной работе многих предприятий. В свою очередь успешная производственная деятельность на каждом отдельном предприятии осуществляется с помощью автоматизированных систем управления производством и предполагает необходимость информационного взаимодействия этих систем. Такое же информационное взаимодействие должно существовать и между отдельными предприятиями при выпуске сложной продукции.

Мехатронная производственная система, представляющая единый комплекс электромеханических, электронных элементов и средств вычислительной техники, между которыми осуществляется постоянный динамически меняющийся обмен энергией и информацией, объединенный общей системой автоматического управления, обладающей элементами искусственного интеллекта, является научной и технической базой для создания современного производства сложной техники и физико-технических систем и процессов различного назначения с принципиально новыми функциями и свойствами.

Для обеспечения согласованной работы мехатронных производственных систем создается соответствующая информационная поддержка между элементами производственной системы по всей цепочке жизненного цикла промышленных изделий:

маркетинг > проектирование > подготовка производства > производство > реализация > эксплуатация > утилизация.

Форма, характер, содержание и назначение информационной поддержки различаются на каждом этапе жизненного цикла изделий. Кроме того, для взаимодействия автоматизированных систем разных предприятий нужна унификация не только формы, но и содержания проектной, технологической, эксплуатационной и другой информации о совместно производимой продукции.

Информационная поддержка сложных мехатронных производств на стадиях проектирования и управления действующими производствами обеспечивается за счет:

- стандартизации структуры проектной, технологической и эксплуатационной документации, а также языков ее представления;

- создания единого информационного пространства (ЕИП) для всех участников жизненного цикла изделия, в котором предусмотрено хранение, обработка и передача информации в объединенных компьютерных средах, а также обеспечен оперативный доступ к данным в нужное время и в нужном месте;

- создания интегрированных компьютеризированных производств и охватывающей их интегрированной автоматизированной информационной системы;

- применения технологии представления данных об изделии (набора методов для представления в электронном виде данных об изделии, относящихся к отдельным процессам жизненного цикла изделия);

- использования автоматизированных систем, называемых системами управления данными об изделии (проектными данными) – PDM (Product Data Management);

- набора методов информационного обеспечения реструктуризации и реинжиниринга производственных процессов (и бизнес-процессов) в целях повышения их эффективности.

В данном курсе лекций основное внимание будет уделено вопросам информационной поддержки методов проектирования и автоматизации управления предприятиями и технологическими процессами.

1.1. Этапы жизненного цикла промышленных изделий

Достижение поставленных целей на современных предприятиях, выпускающих сложные промышленные изделия, оказывается невозможным без широкого использования автоматизированных систем (АС), основанных на применении компьютеров и предназначенных для создания, переработки и использования всей необходимой информации о свойствах изделий и сопровождающих процессов. Специфика задач, решаемых на различных этапах жизненного цикла изделий, обуславливает разнообразие применяемых АС.

Все задачи, решаемые и выполняемые в процессе разработки и производства продукции, называются жизненным циклом (ЖЦ) промышленной продукции (изделий) (product life cycle). Основные этапы жизненного цикла промышленной продукции представлены на рис. 1.1. К ним относятся этапы проектирования, технологической подготовки производства (ТПП), производства, реализации продукции, эксплуатации и утилизации [1].

Этапы жизненного цикла для информационных систем или программного обеспечения (ПО) могут отличаться от приведенных на рис. 1.1.

На всех этапах жизненного цикла изделий имеются свои целевые установки. При этом участники жизненного цикла стремятся достичь поставленных целей с максимальной эффективностью. На этапах проектирования, ТПП и производства нужно обеспечить выполнение технического задания (ТЗ) при заданной степени надежности изделия и минимизации материальных и временных затрат, что необходимо для достижения успеха в конкурентной борьбе в условиях рыночной экономики.

Под эффективностью понимают не только снижение себестоимости продукции и сокращение сроков проектирования и производства, но и обеспечение удобства освоения и снижение затрат на будущую эксплуатацию изделий. Особую важность требование удобства эксплуатации имеет для сложной техники, например в таких отраслях, как авиа- или автомобилестроение.

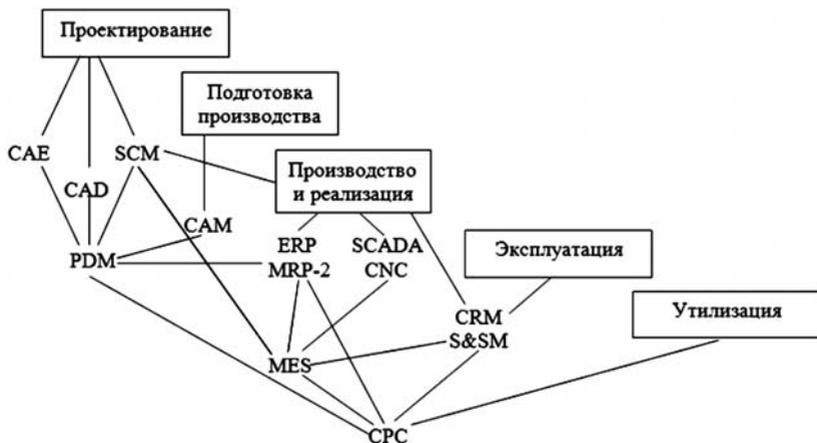


Рис. 1.1. Этапы жизненного цикла промышленных изделий и системы их автоматизации

Основные типы АС с их привязкой к тем или иным этапам жизненного цикла изделий указаны на рис. 1.1:

- CAE – Computer Aided Engineering (автоматизированные расчеты и анализ);
- CAD – Computer Aided Design (автоматизированное проектирование);
- CAM – Computer Aided Manufacturing (автоматизированная технологическая подготовка производства);
- PDM – Product Data Management (управление проектными данными);
- ERP – Enterprise Resource Planning (планирование и управление предприятием);
- MRP-2 – Manufacturing (Material) Requirement Planning (планирование производства);
- MES – Manufacturing Execution System (производственная исполнительная система);
- SCM – Supply Chain Management (управление цепочками поставок);
- CRM – Customer Relationship Management (управление взаимоотношениями с заказчиками);

- SCADA – Supervisory Control and Data Acquisition (диспетчерское управление производственными процессами);
- CNC – Computer Numerical Control (компьютерное числовое управление);
- S&SM – Sales and Service Management (управление продажами и обслуживанием);
- CPC – Collaborative Product Commerce (совместный электронный бизнес);
- PLM – Product Lifecycle Management (управление жизненным циклом изделий).

Современные САПР (или система CAE/CAD), обеспечивающие сквозное проектирование сложных изделий или, по крайней мере, выполняющие большинство проектных процедур, имеют многомодульную структуру. Модули различаются своей ориентацией на те или иные проектные задачи применительно к тем или иным типам устройств и конструкций. При этом возникают естественные проблемы, связанные с построением общих баз данных, с выбором протоколов, форматов данных и интерфейсов разнородных подсистем, с организацией совместного использования модулей при групповой работе. Эти проблемы усугубляются на предприятиях, производящих сложные изделия, в частности, с механическими и радиоэлектронными подсистемами, поскольку САПР машиностроения и радиоэлектроники до недавнего времени развивались самостоятельно, в отрыве друг от друга.

Для решения проблем совместного функционирования компонентов САПР различного назначения разрабатываются системы управления проектными данными – системы PDM. Они либо входят в состав модулей конкретной САПР, либо имеют самостоятельное значение и могут работать совместно с разными САПР.

Уже на этапе проектирования требуются услуги системы SCM, иногда называемой системой управления поставками комплектующих (Component Supplier Management), которая на этапе производства обеспечивает поставки необходимых материалов и комплектующих.

АСТПП, составляющие основу системы САМ, выполняют синтез технологических процессов и программ для оборудования с числовым программным управлением (ЧПУ), выбор технологического оборудования, инструмента, оснастки, расчет норм времени и т. п. Модули систем САМ обычно входят в состав развитых САПР, и потому интегрированные САПР часто называют системами CAE/CAD/CAM/PDM.

Функции управления на промышленных предприятиях выполняются автоматизированными системами на нескольких иерархических уровнях. Автоматизацию управления на верхних уровнях от корпорации (производственных объединений предприятий) до цеха осуществляют АСУП, классифицируемые как системы ERP или MRP-2.

Наиболее известные системы ERP выполняют различные бизнес-функции, связанные с планированием производства, закупками, сбытом продукции, анализом перспектив маркетинга, управлением финансами, персоналом, складским хозяйством, учетом основных фондов и т. п. Системы MRP-2 ориентированы главным образом на бизнес-функции, непосредственно связанные с производством.

АСУТП контролируют и используют данные, характеризующие состояние технологического оборудования и протекание технологических процессов. Именно их чаще всего называют системами промышленной автоматизации.

Для выполнения диспетчерских функций (сбора и обработки данных о состоянии оборудования и технологических процессов) и разработки программного обеспечения для встроенного оборудования в состав АСУТП вводят систему SCADA. Для непосредственного программного управления технологическим оборудованием используют системы CNC на базе контроллеров (специализированных компьютеров, называемых промышленными), встроенных в технологическое оборудование.

На этапе реализации продукции выполняются функции управления отношениями с заказчиками и покупателями, проводится анализ рыночной ситуации, определяются перспективы спроса на планируемые к выпуску изделия. Эти задачи решаются

с помощью системы CRM. Маркетинговые функции иногда возлагаются на систему S&SM, которая, кроме того, служит для решения проблем обслуживания.

На этапе эксплуатации применяются специализированные компьютерные системы, предназначенные для ремонта, контроля, диагностики эксплуатируемых систем. Обслуживающий персонал использует интерактивные учебные пособия и технические руководства, а также средства для дистанционного консультирования при поиске неисправностей, программы для автоматизированного заказа деталей взамен отказавших.

Следует отметить, что функции некоторых автоматизированных систем часто перекрываются. В частности, это относится к системам ERP и MRP-2. Управление маркетингом может быть поручено как системе ERP, так и системе CRM или S&SM.

На решение оперативных задач управления проектированием, производством и маркетингом ориентированы системы MES. Они близки по некоторым выполняемым функциям к системам ERP, PDM, SCM, S&SM и отличаются от них именно оперативностью, принятием решений в реальном времени, причем большое значение придается оптимизации этих решений с учетом текущей информации о состоянии оборудования и процессов.

Перечисленные автоматизированные системы могут работать автономно, что в настоящее время обычно и происходит. Однако эффективность автоматизации будет заметно выше, если данные, генерируемые в одной из систем, будут доступны в других системах, поскольку принимаемые в них решения станут более обоснованными.

Чтобы достичь должного уровня взаимодействия промышленных автоматизированных систем, требуется создание единого информационного пространства не только на отдельных предприятиях, но и, что более важно, в рамках объединения предприятий (системы CPC или PLM – Product Lifecycle Management).

Единое информационное пространство обеспечивается благодаря унификации как формы, так и содержания информации о конкретных изделиях на различных этапах их жизненного цикла. Унификация формы достигается использованием стандартных

форматов и языков представления информации в межпрограммных обменах и при документировании, т. е. применением так называемых CALS (Continuous Acquisition and Lifecycle Support) – технологий, которые будут рассмотрены ниже.

Более подробно основные функции, проектные процедуры и программное обеспечение АС различных этапов жизненного цикла промышленных изделий (см. рис. 1.1) будут описаны в последующих лекциях.

1.2. Системный подход к информационной поддержке сложных систем: анализ, синтез, моделирование, оптимизация [2, 3]

Системный подход

Системный подход является общим подходом, выражающим основные идеи и принципы, используемые при создании и управлении сложными техническими системами. Системный подход включает в себя самостоятельные математические дисциплины, такие как «Системный анализ» и «Теория систем».

В широком смысле системный анализ – это дисциплина, занимающаяся проблемами принятия решений, когда выбор альтернативы требует анализа сложной информации различной физической природы. Системный анализ возник в связи с необходимостью вести исследования междисциплинарного характера при создании сложных систем (технических, экономических, экологических). Основы системного анализа лежат в дисциплинах, занимающихся проблемами принятия решений, – теории исследования операций и общей теории управления.

В узком смысле системный анализ включает в себя выявление структуры системы, типизацию связей (группировка связей по характерным принципам), определение свойств (атрибутов) системы, анализ влияния внешней среды и разработку системы управления.

В отличие от системного анализа теория систем носит прикладной характер, ориентированный на решение конкретных прикладных задач.

В технике дисциплину, в которой исследуются сложные технические системы и их проектирование, чаще всего называют «Системотехникой» (вместо «Теория систем» и «Системный анализ»). Предметом системотехники являются, во-первых, организация процесса создания, использования и развития технических систем, а во-вторых, методы их проектирования и исследования.

Следуя [3], рассмотрим ряд основных понятий и терминов системотехники.

Система – множество элементов, находящихся в отношениях и связях между собой.

Элемент – такая часть системы, представление о которой целесообразно подвергать при проектировании дальнейшему членению.

Сложная система – система, характеризующаяся большим числом элементов и, что наиболее важно, большим числом взаимосвязей элементов. Сложность системы определяется также видом взаимосвязей элементов, свойствами целенаправленности, целостности, членимости, иерархичности, многоаспектности. Очевидно, что современные автоматизированные информационные системы и, в частности, САПР являются сложными в силу наличия у них перечисленных свойств и признаков.

Подсистема – часть системы (подмножество элементов и их взаимосвязей), которая имеет свойства системы.

Надсистема – система, по отношению к которой рассматриваемая система является подсистемой.

Структура – отображение совокупности элементов системы и их взаимосвязей; понятие структуры отличается от понятия самой системы также тем, что при описании структуры принимают во внимание лишь типы элементов и связей без конкретизации значений их параметров.

Параметр – величина, выражающая свойство или системы, или ее части, или влияющей на систему среды. Обычно в моделях

систем в качестве параметров рассматривают величины, не изменяющиеся в процессе исследования системы. Параметры подразделяют на внешние, внутренние и выходные, выражающие свойства элементов системы, самой системы, внешней среды соответственно. Векторы внутренних, выходных и внешних параметров далее обозначены $X = (x_1, x_2, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_m)$, $Q = (q_1, q_2, \dots, q_k)$ соответственно.

Фазовая переменная – величина, характеризующая энергетическое или информационное наполнение элемента или подсистемы.

Состояние – совокупность значений фазовых переменных, зафиксированных в одной временной точке процесса функционирования (событие).

Поведение (динамика) системы – изменение состояния системы в процессе функционирования.

Система без последствия – ее поведение при $t > t_0$ определяется заданием состояния в момент t_0 и вектором внешних воздействий $Q(t)$. В системах с последствием, кроме того, нужно знать предысторию поведения, т.е. состояния системы в моменты, предшествующие t_0 .

Вектор переменных V , характеризующих состояние (вектор переменных состояния), – избыточное множество фазовых переменных, задание значений которых в некоторый момент времени полностью определяет поведение системы в дальнейшем (в автономных системах без последствия).

Пространство состояний – множество возможных значений вектора переменных состояния.

Фазовая траектория – представление процесса (зависимости $V(t)$) в виде последовательности точек в пространстве состояний.

К характеристикам сложных систем, как сказано выше, часто относят следующие понятия.

Целенаправленность – свойство искусственной системы, выражающее назначение системы. Это свойство необходимо для оценки эффективности вариантов системы.

Целостность – свойство системы, характеризующее взаимосвязанность элементов и наличие зависимости выходных пара-

метров от параметров элементов, при этом большинство выходных параметров не является простым повторением или суммой параметров элементов.

Иерархичность – свойство сложной системы, выражающее возможность и целесообразность ее иерархического описания, т. е. представления в виде нескольких уровней, между компонентами которых имеются отношения целое-часть.

В качестве примера сложной системы рассмотрим «Компьютер».

Компьютер является сложной системой в силу наличия у него большого числа элементов, разнообразных связей между элементами и подсистемами, свойств целенаправленности, целостности, иерархичности. К подсистемам компьютера относятся процессор (процессоры), оперативная память, кэш-память, шины, устройства ввода-вывода. В качестве надсистемы могут выступать вычислительная сеть, автоматизированная и (или) организационная система, к которым принадлежит компьютер. Внутренние параметры – времена выполнения арифметических операций, чтения (записи) в накопителях, пропускная способность шин и др. Выходные параметры – производительность компьютера, емкость оперативной и внешней памяти, себестоимость, время наработки на отказ и др. Внешние параметры – напряжение питания сети и его стабильность, температура окружающей среды и др.

Анализ, синтез, моделирование и оптимизация систем

Изучение сложных систем опирается на методы анализа, синтеза, моделирования и оптимизации систем.

Составными частями системотехники являются следующие основные разделы:

- иерархическая структура системы;
- анализ и моделирование систем;
- синтез и оптимизация систем.

Необходимость иерархической организации в технических системах – следствие их сложности, когда централизованная

обработка информации либо просто невозможна, либо требует такой затраты времени (или средств), которая недопустима по техническим условиям. Основная задача иерархической организации – распределение функций обработки информации и принятия решений между отдельными элементами системы.

Представление о производственной системе расчлняют на иерархические уровни. На верхнем уровне используются только самые общие черты и особенности производственной системы. На следующих – степень подробности описания возрастает (при этом рассматриваются уже отдельные блоки системы с учетом их взаимодействия), что позволяет на каждом иерархическом уровне формулировать задачи приемлемой сложности.

Для большинства приложений характерны следующие иерархические уровни:

- **системный уровень**, на котором решаются наиболее общие задачи проектирования систем, машин и процессов; результаты проектирования представляют в виде структурных схем, генеральных планов, схем размещения оборудования и т. д.;

- **макроуровень**, на котором исследуются отдельные устройства, узлы машин и приборов; результаты представляют в виде функциональных, принципиальных и кинематических схем, сборочных чертежей и т. д.;

- **микроуровень**, на котором проектируют отдельные детали и элементы машин и приборов.

В каждом приложении число выделяемых уровней и их наименований может быть различным (в радиотехнике, вычислительной технике, машиностроении и т. п.). В машиностроении имеются уровни деталей, узлов, машин и комплексов.

В широком смысле **анализ** – это изучение конкретных фактов, вскрытие тонкой структуры явления (системы). **Синтез** – объединение различных факторов, построение структуры системы, развитие того или иного процесса, установление его связи с другими явлениями с учетом их взаимной обусловленности.

Основная задача научного анализа – выделить реальные взаимодействия (зависимости) материальных объектов из

множества мысленно допустимых и сформулировать принципы их отбора. Описание этих принципов отбора (в терминах и переменных, которые наиболее полно характеризуют изучаемый предмет) является предметом **моделирования** процессов (систем).

Моделирование имеет две четкие различимые задачи:

- создание моделей сложных систем (в англоязычном написании – modelind);
- анализ свойств систем на основе исследования их моделей (simulation).

Моделирование можно разделить на аналитическое и физическое. К аналитическому моделированию относятся математическое и имитационное моделирование, геометрическое, а также виртуальное моделирование. Физическое моделирование реализует физическую модель объекта (например, технология быстрого прототипирования – rapid prototyping).

В данной лекции рассмотрим только математическое и имитационное моделирование. Остальные виды моделирования будут рассмотрены в последующих лекциях.

Построение математических моделей является основой всего системного анализа. Это – центральный этап исследования или проектирования любой системы. От качества модели зависит судьба всего последующего анализа.

В качестве примера рассмотрим модель, которая используется для оптимизации некоторых действий.

Модель динамического процесса описывается дифференциальным уравнением вида

$$\dot{x} = f(x, t, u), \quad (1.1)$$

где вектор функции $U(t, x)$ называется управлением;

$x = (x_1, x_2, \dots, x_n)$ – вектор состояния системы.

Управление $U(t, x)$ выбирается из условия достижения некоторой цели. Весьма распространенный класс задач с помощью модели (1.1) можно описать следующим образом: за время T перевести систему из состояния

$$x(0) = x_0 \quad (1.2)$$

в состоянии

$$x(T) = x_T \quad (1.3)$$

так, чтобы целевая функция $F(x,t,u)$ была минимальной, т. е.

$$\int_0^T F(x,t,u) dt \rightarrow \min. \quad (1.4)$$

Ограничения (1.2), (1.3) и целевую функцию (1.4) мы не включаем в понятие модели (1.1) (для одной и той же модели могут ставиться разные задачи).

Потребность и точность выбираемой или составляемой математической модели должны находиться в соответствии с требуемым качеством решения задачи. Излишняя подробность модели приводит к неоправданному увеличению машинного времени, недостающая – к неправильному результату. Весьма желательна проверка адекватности принятой модели.

Применение математических моделей для исследования и проектирования сложных систем связано с рядом принципиальных трудностей:

- отсутствие законченной постановки задачи исследования при незаконченном процессе познания объекта моделирования;
- сложные и трудные математические процедуры применения аналитических методов решения проблемных задач;
- необходимость изучать новые ситуации в сложных системах, о которых мало что известно;
- невозможность введения в изучаемую систему новых компонентов и т. д.

Для решения подобных задач мало наличие доброкачественной модели, необходима также надежная исходная информация. Но и информации еще не достаточно. Необходим определенный сервис. Так, система моделей должна быть доступна исследователю, варианты расчетов должны проходить достаточно быстро, должна хорошо функционировать система визуализации результатов. Должен быть предельно облегчен ввод новой информации, переход к новым вариантам расчета и т. д. Другими словами,

должна быть создана специальная «имитационная система». Отсюда следует, что «имитационная система – это совокупность моделей, имитирующих протекание изучаемого процесса, объединенная со специальной системой вспомогательных программ и информационной базой, позволяющих достаточно просто и оперативно развивать варианты расчетов. Имитационной системе присуща определенная архитектура, и она должна быть снабжена четкими продуктами ее использования» [2]. Имитационные системы могут быть успешно использованы в сложных задачах оптимизации.

Имитационное моделирование (ИМ) предполагает проведение численных экспериментов с математическими моделями, описывающими поведение систем, с привлечением для этого ЭВМ. Таким образом, имитационная модель представляет собой машинный аналог сложной системы, она позволяет заменить эксперимент с реальной системой экспериментом с математической моделью этой системы в ЭВМ. Поэтому ИМ так широко применяется для проектирования сложных объектов и для изучения сложных систем, особенно в тех случаях, когда реальный эксперимент слишком дорог или вообще невозможен.

Имитационное моделирование позволяет использовать как точную количественную информацию, так и информацию, полученную от экспертов с учетом оценок, суждений и различных эвристик. В этом состоит главное преимущество ИМ перед математическими моделями.

Имитационную модель сложной системы (СС) можно использовать при решении задач в следующих случаях:

- если не существует законченной постановки задачи исследования – идет процесс познания объекта моделирования; имитационная модель служит средством изучения явления;

- если аналитические методы имеются, но математические процедуры столь сложны и трудоемки, то имитационное моделирование дает более простой способ решения задачи;

- когда кроме оценки влияния параметров СС желательно наблюдение за поведением компонентов СС в течение определенного периода;

- если имитационное моделирование оказывается единственным способом исследования сложной системы из-за невозможности наблюдения явлений в реальных условиях;

- когда необходимо контролировать протекание процессов в СС путем замедления или ускорения явлений в ходе имитации;

- при подготовке специалистов и освоении новой техники, когда на имитационной модели обеспечивается возможность приобретения необходимых навыков в эксплуатации новой техники;

- когда изучаются новые ситуации в СС, о которых мало что известно или не известно ничего; в этом случае имитация служит для предварительной проверки новых стратегий и правил принятия решений перед проведением экспериментов на реальной системе;

- когда особое значение имеет последовательность событий в проектируемой СС и модель используется для предсказания «узких» мест в функционировании системы и других трудностей, появляющихся в поведении СС при введении в нее новых компонентов.

Недостатки ИМ: разработка хорошего ИМ часто обходится дороже создания математической модели и требует больших временных затрат; ИМ не может точно отражать реальную сложную систему (быть полностью адекватной сложной системе).

На рис. 1.2 показаны основные этапы имитационного моделирования.

Создание системы начинается с выбора ее структуры. Под структурой в общем случае принято понимать вид элементов, из которых состоит объект, и отношений между элементами. Так, завод как объект состоит из цехов, отделов, служб и т. д. Структура объекта может быть различной в зависимости от назначения объекта.

Синтез объекта (производственного процесса) в значительной степени определяет потребительские свойства будущей продукции. Следует отметить, что анализ и синтез непрерывно связаны друг с другом, так как анализ позволяет получить необходимую информацию для целенаправленного выполнения процедур синтеза в итерационном процессе разработки производственного процесса.

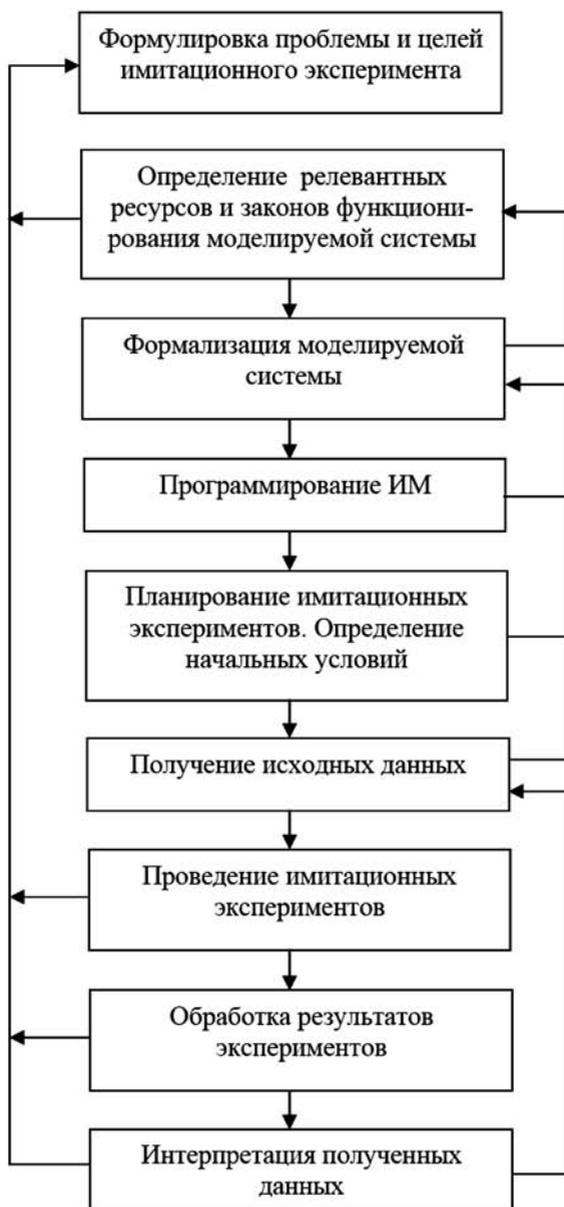


Рис. 1.2. Этапы имитационного моделирования

Синтез, как и анализ, подразделяется на две задачи:

- синтез структуры производственных процессов (систем) – *структурный синтез*;
- выбор численных значений параметров элементов систем – *параметрический синтез*.

Задача структурного синтеза формулируется в системотехнике как задача принятия решения (ЗПР). Ее суть заключается в выборе решения среди множества возможных решений и ограничивающих условий. Принятие решения охватывает широкий круг задач и процедур – от выбора вариантов в конечных и обозримых множествах до задач творческого характера, не имеющих формальных способов решения.

Задачи синтеза структур объектов относятся к наиболее трудно формализуемым. Существует ряд общих подходов к постановке этих задач, однако практическая реализация большинства из них не очевидна. Именно по этой причине структурный синтез, как правило, выполняют в интерактивном режиме при решающей роли инженера-разработчика, а ЭВМ играет вспомогательную роль: выполнение отдельных расчетов, предоставление необходимых справочных данных, фиксация промежуточных и окончательных данных. Имеются лишь отдельные примеры успешной автоматизации структурного синтеза технологических процессов и управляющих программ для механообработки в машиностроении, проектирования печатных плат и кристаллов БИС и некоторые другие. В целом постановки и методы решения задач структурного синтеза в связи с трудностями формализации не достигли степени обобщения и детализации, свойственной математическому обеспечению процессов анализа. Достигнутая степень обобщения выражается в установленной типичной последовательности действий и видов описания при их использовании в САПР. Большое значение для развития подсистем синтеза в САПР имеют разработка и унификация языков представления описаний (спецификаций).

Создание системы (производственного процесса) всегда начинается со *структурного синтеза*, который определяет общий облик будущей системы (одну из конструкций механизма, схему

технологического процесса). Но эти конструкции и схемы выбирают в параметрическом виде, т. е. без указания числовых значений параметров элементов. Поэтому нужно задать или рассчитать значения этих параметров, т. е. выполнить *параметрический синтез*. Примерами результатов параметрического синтеза могут служить, например, рабочие режимы (параметры) технологических операций производственного процесса.

В случае если по результатам анализа проектное решение признается неокончательным, то начинается процесс последовательных приближений к приемлемому варианту проекта. Во многих приложениях для этого удобно варьировать значения параметров элементов, т. е. использовать параметрический синтез на базе многовариантного анализа. При этом задача параметрического синтеза при неизменной структуре объекта может быть сформулирована как задача параметрической оптимизации (или просто оптимизации). Если параметрический синтез не приводит к успеху, то повторяют процедуру структурного синтеза (корректируют или перебирают структуру объекта).

Следуя [3], рассмотрим постановку задач параметрического и структурного синтеза.

Базовая задача параметрического синтеза (приведенная к задаче оптимизации) ставится как задача математического программирования:

$$\begin{aligned} & \text{extr } F(X), \\ & X \in Dx \\ & Dx = \{X / y(X) > 0, \psi(X) = 0\}, \end{aligned} \tag{1.5}$$

где $F(X)$ – целевая функция; X – вектор управляемых (проектных) параметров; $y(X)$ и $\psi(X)$ – функции ограничения; Dx – допустимая область в пространстве управляемых параметров.

Запись (1.5) интерпретируется как задача поиска экстремума целевой функции путем варьирования управляемых параметров в пределах допустимой области. Таким образом, для выполнения расчета номинальных значений параметров необходимо,

во-первых, сформулировать задачу в виде (1.5), во-вторых, решить задачу поиска экстремума $F(X)$.

Базовая задача структурного синтеза формулируется как *задача принятия решения* (ЗПР) следующим образом:

$$\text{ЗПР} = \langle A, Y, \text{Мод.}, \Pi \rangle, \quad (1.6)$$

где A – множество альтернатив проектного решения; $Y = (y_1, y_2, \dots, y_m)$ – множество критериев (выходных параметров), по которым оценивается соответствие альтернативы поставленным целям; $\text{Мод.}: A \rightarrow Y$ – модель, позволяющая для каждой альтернативы рассчитать вектор критериев; Π – решающее правило для выбора наиболее подходящей альтернативы в многокритериальной ситуации.

В свою очередь, каждой альтернативе конкретного приложения можно поставить в соответствие значение упорядоченного множества (набора) атрибутов $X = x_1, x_2, \dots, x_n$, характеризующих свойство альтернативы. Множество X называют *записью* (в теории баз данных), *фреймом* (в искусственном интеллекте) или *хромосомой* (в генетических алгоритмах).

Во многих случаях удается построить в виде алгоритма математическую модель, позволяющую по заданному множеству X и заданной структуре объекта рассчитать вектор критериев Y . Такие модели получаются автоматически в программах анализа типа Spice Adams или ПА-9 для объектов, исследуемых на макроуровне. Для построения таких моделей можно использовать методы планирования экспериментов или подход, основанный на использовании искусственных нейронных сетей.

Если же математическая модель $A \rightarrow Y$ остается неизвестной, то ее создают на основе *экспертных систем*. В [3] описано большое число методов, входящих в состав математического обеспечения анализа и синтеза проектных решений.

Математический аппарат для каждого из иерархических уровней проектирования имеет свои особенности. На *микроуровне* типичные математические модели представлены дифференциальными уравнениями в частных производных вместе с краевыми условиями (такие модели называются *распределенными*).

Объектами исследования здесь являются поля различных физических величин (напряжений, температур, скоростей и т. д.).

Моделями *макроуровня* (*сосредоточенными* моделями) являются системы алгебраических и обыкновенных дифференциальных уравнений.

Моделями *системного* уровня моделирования сложных объектов (типа сложных производственных систем) является аппарат теории массового обслуживания, сетей Петри или имитационного моделирования.

Постановки и методы решения задач *структурного синтеза* относятся к наиболее трудно формализуемым и выполняются, как правило, в интерактивном режиме инженером-разработчиком. Процесс формального описания структурного синтеза (как было отмечено выше) часто сводится к задаче принятия решения или генетическим методам и алгоритмам.

Процедуры *параметрического синтеза* выполняются либо человеком в процессе многовариантного анализа (в интерактивном режиме), либо решаются на базе формальных методов оптимизации (как было описано выше). В последнем случае находят применение несколько постановок задач оптимизации в виде (1.5).

Последние достижения анализа и синтеза сложных производственных систем, базирующихся на принципах использования современных информационных технологий и программных комплексов, будут рассмотрены в последующих лекциях, посвященных проектированию и управлению производственными процессами.

Контрольные вопросы

1. Форма, характер, содержание и назначение информационной поддержки мехатронных производств.
2. Дайте характеристику этапов жизненного цикла промышленных изделий.
3. Назовите основные типы автоматизированных систем (АС) с их привязкой к тем или иным этапам жизненного цикла изделий.
4. Дайте определение «сквозного проектирования» сложных изделий и возникающих при этом проблем.

5. Дайте краткое определение АСТПП, АСУП и АСУТП.
6. Дайте определение понятия «Системный подход» в широком и узком смыслах.
7. Опишите основные понятия и термины системотехники (не цитируя дословно текст лекции).
8. Назовите признаки, присущие сложной системе, приведите собственный пример сложной системы.
9. Обоснуйте необходимость иерархической организации в сложных технических системах, укажите иерархические уровни производственных систем.
10. Укажите основные части (разделы) системотехники, назначение и задачи анализа и моделирования систем.
11. Опишите суть (основное содержание) аналитического и физического моделирования.
12. Обоснуйте необходимость применения математических моделей для исследования и проектирования сложных систем (принципиальные трудности моделирования).
13. Укажите требования, предъявляемые к математическим моделям (на примере имитационной модели).
14. Опишите основное назначение и область применения имитационного моделирования.
15. Основная задача синтеза и оптимизации сложных систем (структурный и параметрический синтез).
16. Порядок создания сложных систем (производственных процессов), постановка задачи параметрического и структурного синтеза.

Лекция 2

МАШИНОСТРОИТЕЛЬНЫЕ И МЕТАЛЛООБРАБАТЫВАЮЩИЕ ПРОИЗВОДСТВА

Металлургия, химическая промышленность, машиностроение и металлообработка являются взаимосвязанными базовыми производствами обрабатывающей отрасли промышленности. Машиностроение во многом определяет состояние производственного потенциала страны, являясь базой технического перевооружения всего общественного производства. Удельный вес продукции машиностроения и металлообработки в общем объеме промышленного производства в России составляет около 20 %.

Основными видами промышленного производства являются следующие производства: одиночное, мелкосерийное, серийное и крупносерийное, массовое, а также непрерывное и дискретное. Под дискретным производством понимается изготовление продукта, проходящего через конечное число технологических и сборочных операций. Непрерывное производство подразумевает изготовление продукта, претерпевающего непрерывные изменения, в результате которых заготовка превращается в готовую деталь. Вид производства оказывает существенное влияние на структуру, систему управления и стоимость продукции (эффективность), технологических процессов данных производств. Кроме того, вид производства определяет необходимый уровень гибкости производства. Под гибкостью производства понимается способность быстро и без существенных затрат труда и средств перестраиваться на изготовление новой или модернизированной продукции и на новые технологические процессы с новой их организацией. По сути дела, речь идет о новых организационных формах решения стратегических задач развития производства, его интенсификации

при ограниченных трудовых, материальных и энергетических ресурсах. Решения таких задач возможно только при условии полной автоматизации машиностроительного производства за счет создания комплексно автоматизированных производств, эффективность которых в первую очередь зависит от гибкости и степени интегрированности их структурных составляющих.

Высшей формой развития комплексно-автоматизированных производств являются гибкие производственные системы (ГПС) и гибкие автоматизированные производства (ГАП) [4].

2.1. Гибкие автоматизированные системы и производства

Такие производства позволяют быстро переходить на выпуск новой продукции, осуществлять ее модернизацию, совершенствовать технологию производства, прежде всего путем смены управляющих программ. Антиподом ГПС и ГАП в отношении гибкости являются автоматические линии (АЛ) с так называемой жесткой автоматизацией. Эффективность АЛ и ГПС зависит от видов производств (рис. 2.1).



Рис. 2.1. Зависимость стоимости продукции от объема ее выпуска для различных видов производства [5]

Типовой состав ГАП в наиболее полном виде состоит из двух частей, приведенных на рис. 2.2:

- гибкая производственная система (ГПС), которая непосредственно реализует технологический процесс изготовления изделий;
- автоматизированные системы проектирования, подлежащих изготовлению изделий и технологической подготовки их производства, реализуемые с помощью показанных в нижней части рисунка автоматизированных рабочих мест (АРМ).

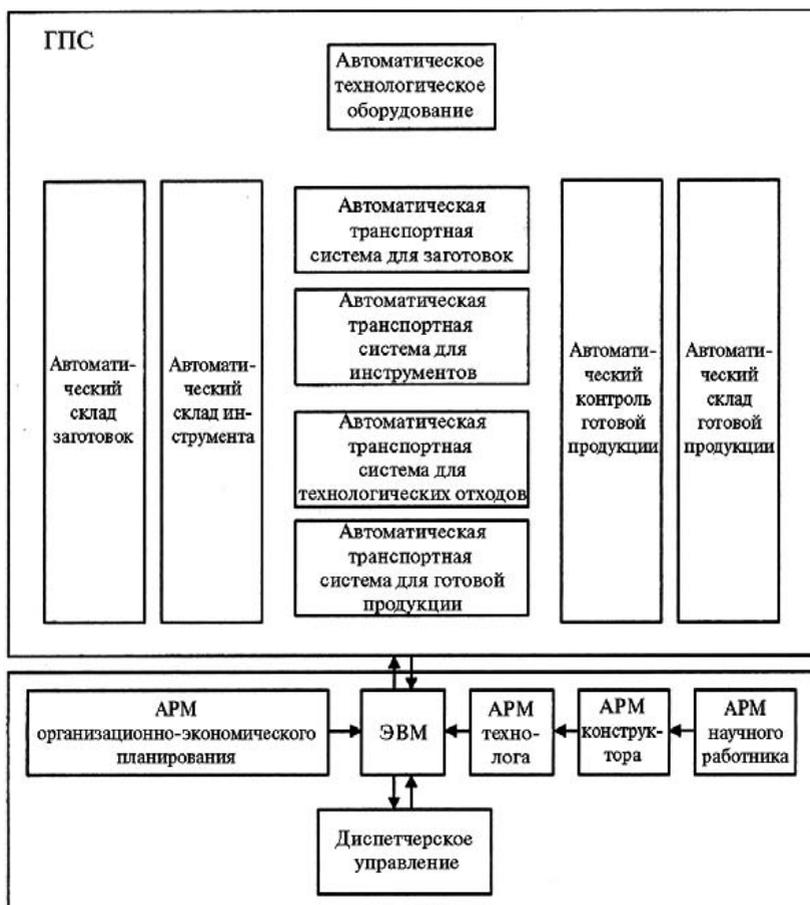


Рис. 2.2. Состав гибкого автоматизированного производства [5]

Части объединены общей автоматизированной системой управления (АСУ) на базе локальной компьютерной сети.

Предел, к которому стремится процесс гибкой автоматизации производства, можно кратко определить так:

- автоматическое производство изделий сколь угодно малыми партиями;
- себестоимость и производительность, близкие к достигнутым в современном массовом производстве;
- практически безлюдное производство – количество работающих, по сравнению с существующим, меньше на два порядка;
- комплексная автоматизация всех частей производства, включая технологические процессы, подготовку производства, разработку документации на выпускаемые изделия, планирование и управление производством в целом.

Таким образом, в максимально полном исполнении ГАП на его вход поступают технические задания на очередное изделие, а на выходе – готовая продукция. Весь процесс может происходить непрерывно на основе машинных носителей информации, т. е. без бумажной документации.

Для мелкосерийного производства степень его гибкости характеризуется количеством типов изделий, которые могут изготавливаться путем смены только программ. Типичные значения этой величины для современных ГАП металлообработки – десятки типов деталей и сотни их модификаций.

В мире внедрение гибкой автоматизации связывают со следующими конкретными достоинствами:

- повышение конкурентоспособности продукции (за счет более быстрой сменяемости выпускаемых изделий, повышения их качества, экономии ресурсов и труда);
- рост производительности;
- стабилизация производства и ускорение оборачиваемости капитала;
- повышение эффективности использования рабочей силы;
- создание новых технологических процессов и производств.

Основной принцип построения гибких производств – компоновка их из комплектно поставляемых унифицированных модулей. В соответствии с составом ГАП, показанным на рис. 2.2,

основными такими модулями являются технологические модули-ячейки, модули транспортные (в том числе на базе транспортных навесных и напольных роботов), складские (в том числе с роботами-штамбелерами), модули контроля качества продукции и технической диагностики оборудования. Последние модули образуют систему автоматического контроля качества, без которой невозможна автоматическая работа всего комплекса. В задачи таких систем входят собственно измерение контролируемых параметров, регистрация результатов этих измерений, визуализация их на терминалах, подача управляющих воздействий в систему управления оборудованием, когда по этим параметрам необходима автоматическая корректировка технологических процессов (например, в случае адаптивного управления станками, учитывающего износ инструмента, изменение размеров заготовок и т.п.). Примером оборудования, используемого в системах автоматического контроля ГАП, являются робототехнические контрольно-измерительные машины.

В ГПС, входящих в состав ГАП (см. рис. 2.2), используют прогрессивное оборудование для перемещения материала, управления станками, формообразования, а также для перспективных методов обработки. Степень необходимой гибкости системы зависит главным образом от объема производства и разнообразия деталей. Так, фирма Comau (США) изготавливает станки для обработки 3...40 деталей в час; на одной линии обрабатывается до 40 разнообразных деталей.

ГПС позволяют преодолеть трудности, связанные с производством групп деталей среднего объема. Эти системы объединяют станки с ЧПУ и автоматическое оборудование для перемещения материала. Система в целом и сам процесс обработки управляются с ЭВМ.

ГПС состоит из нескольких обрабатывающих модулей с ЧПУ; средств для транспортировки палет, размещения зажимов и хранения на участке; централизованного устройства управления движением палет и обрабатывающих операций; различных систем для контроля, наладки инструмента и коррекции, удаления стружки, промывки и продувки.

ГПС, по сравнению со специализированными станочными линиями, легко настраиваются на изменение номенклатуры деталей; новые детали можно обрабатывать при минимальных затратах времени и стоимости необходимого специализированного оборудования.

Основные преимущества ГПС – мгновенная реакция на потребности рынка сбыта, снижение затрат на производство деталей за счет сокращения времени, экономии площади и уменьшения капиталовложений, а также экономия за счет отказа от поступления запчастей извне. Способность обрабатывать детали различной конфигурации позволяет изготовителю осуществлять массовое производство за счет сочетания операций при изготовлении малых и средних партий деталей. Возможность обработки деталей произвольного ассортимента исключает необходимость складирования деталей в ожидании переналадки системы, в результате чего снижаются запасы промежуточных продуктов, потребность в площадях и обеспечивается гибкость планирования производства. Кроме того, к преимуществам ГПС следует отнести и то, что если один участок (или модуль с ЧУ) имеет отклонение от заданного режима работы, то это не влечет за собой отключения всей системы, так как операции неработоспособного модуля распределяются между оставшимися роботами. В результате темп производства снижается, но оно не останавливается. При использовании ГПС имеется возможность экономии производственных площадей за счет исключения специализированного оборудования, сокращения потребности в рабочей силе и снижения степени зависимости от операторов, улучшения условий труда и техники безопасности.

К преимуществам ГПС, по сравнению с обрабатывающими центрами, можно отнести:

- повышение коэффициента использования станков и возможность использования высокопроизводительных рабочих устройств (многошпиндельных головок);
- управление процессами от ЭВМ, обеспечивающее максимальную производительность или максимальный ассортимент обрабатываемых деталей;

- сокращение производственных площадей на 25...30 %;
- сокращение заделов (для некоторых операций и полное их исключение);
- возможность улучшения условий труда, снижение шума.

Чтобы обеспечить оборудованием производство групп деталей без дополнительных капиталовложений (на приобретение специализированного оборудования), фирма Somaу запатентовала ГПС, основанную на следующих концепциях использования:

- специальные автономные модули ЧПУ, каждый из которых предназначен для конкретного типа обработки. Специализированный характер таких модулей снижает капиталовложения, упрощает и оптимизирует обработку (обрабатывающий центр, наоборот, представляет собой технический компромисс);
- модульный транспорт между станками;
- стандартизированные палеты; базирование и зажим специальными приспособлениями;
- оборудование модульной конструкции.

Примерами станочных модулей с ЧПУ являются фрезерный модуль MILL, модуль для сверления и нарезания резьбы метчиком DRILL-INDEX, модуль смены многшпindelной головки MULTIFLEX (MFX), многоцелевой модуль MULTI-STORE (MSR), многоцелевой модуль с наклоняемым шпинделем MULTI-STORE-TILTING, токарный модуль TURN-INDEX.

Фрезерный модуль MILL имеет горизонтально обрабатывающее устройство с ЧПУ, шпиндель и подвижную стойку; движение по осям X, Y и Z обеспечивается антифрикционными направляющими плоского типа, приводимыми в действие серводвигателями; точность позиционирования по осям X и Y составляет $\pm 0,05$ мм по всей длине перемещения (для оси Z $\pm 0,03$ мм). Фрезерная головка приводится в действие двигателем постоянного тока мощностью 1,5 кВт через двухскоростную коробку передач.

Модуль для сверления и нарезания резьбы содержит индивидуальный привод для шестипозиционной горизонтальной поворотной головки. Стойка движется по осям X и Y, перемещение по осям X, Y и Z обеспечивается плоскими направляющими, приводимыми в действие серводвигателем. Точность позиционирования

по осям X, Y и Z $\pm 0,05$ мм по всей длине хода. Шестипозиционная поворотная револьверная головка приводится в действие двигателем с постоянным крутящим моментом; двигатель постоянного тока мощностью 11 кВт (имеется три скорости). Модуль оснащен инфракрасным датчиком для контроля инструмента.

Модуль TAP-INDEX оснащен шестипозиционной поворотной револьверной головкой, установленной на подвижной стойке. Время смены инструмента 3 с. Точность позиционирования осей X, Y и Z $\pm 0,1$ мм. Двигатель постоянного тока с постоянным крутящим моментом мощностью 5 кВт приводит в действие горизонтальный шпиндель, стандартная коробка передач обеспечивает две скорости. Инфракрасный датчик контролирует износ инструмента.

Модуль смены многшпиндельной головки MULTI-FLEX состоит из устройства для горизонтальной обработки с направляющими и ЧПУ; мощность привода шпинделя 11 кВт. На передней поверхности направляющих расположены установочные и зажимные устройства, а также устройство соединения привода шпиндельной головки. Горизонтальные механизмы барабанного типа емкостью до 6 головок располагаются по обе стороны направляющего устройства. При работе направляющих контроллер ЧПУ подает команду следующей (по последовательности операций) головке для поворота в положение загрузки. Второй барабан получает команду на поворот свободного гнезда в положение разгрузки. При возврате направляющих обе головки (в рабочем положении и положении ожидания) разжимаются и двигаются горизонтально. Затем головки зажимаются, приводится в действие привод, и цикл повторяется. Важно отметить, что головки из магазина могут использоваться произвольно, что позволит главной ЭВМ оптимизировать рабочую загрузку системы в целом. Модуль может использовать щуп; если используется поворотный стол и палета, которые могут поворачиваться на 180 °С, то зондирование производится щупом с несколькими головками, управляемыми от ЭВМ. Размер головки 600 × 600 мм.

Многоцелевой модуль MULTI-STORE состоит из обрабатывающего модуля с ЧПУ с подвижной стойкой и горизонтального

шпинделя с гидравлической балансировкой. Имеется автоматическая система смены инструмента с магазином емкостью 30 инструментов; время смены от 6 до 9 с. Движение по осям X, Y и Z осуществляется с помощью роликовых направляющих плоского типа, точность позиционирования $\pm 0,015$ мм по всей длине перемещения обеспечивается прецизионными шариковыми парами. Система управления позиционированием использует позиционный преобразователь типа индуктосина, который после фиксирования оси зажимается во избежание сдвига. Шпиндель приводится в действие двигателем постоянного тока мощностью 11 кВт с коробкой передач на 2 скорости. Постоянная температура обеспечивается масляной системой охлаждения.

Транспортная система. В каждой ГПС существует, по крайней мере, один загрузочно-разгрузочный участок. В зависимости от вида обработки детали это может быть загрузочно-перегрузочно-разгрузочный участок, который включает в себя автоматические электромеханические затяжные устройства (горизонтальные и вертикальные), зажимающие деталь в палете; верхний специальный зажим (универсальный для большей части деталей); нижнюю стандартную часть палеты (общую для всех деталей стандартного перемещения). Все палеты транспортируются через систему с помощью рольганга модульного исполнения с цепным приводом. На каждом рабочем участке палеты устанавливаются, идентифицируются и получают от ЭВМ команду о направлении дальнейшего движения. Если участок занят или на нем выполняется операция, ЭВМ подает команду палете о перемещении к другому модулю или об обратном движении и возвращении на этот участок через какой-то период времени. Гидравлическая перемещающаяся штанга подает палету к возвращающему столу обрабатывающего модуля или отводит от него.

Контроль и управление ГПС. ГПС обычно имеет два основных уровня ЧПУ. При каждом обрабатывающем модуле система ЧПУ типа CNC или PCNC подает на станок команду о выборе требуемого инструмента с учетом последовательности обработки и вспомогательных функций. Второй уровень включает одну или две ЭВМ-супервизора. На этом уровне обеспечивается

координация всех модулей в системе. При введении в систему для обработки новой детали ЭВМ-супервизор проверяет наличие соответствующей программы по этой детали и наличие требуемых инструментов; если ЭВМ определяет, что система не может обрабатывать новую деталь, то деталь и ее палета не получают доступа в систему.

После установки детали в систему ЭВМ-супервизор оптимизирует прохождение детали по системе. Палета останавливается и опознается на каждом участке. Если модуль на ее пути занят или не работает, палета получает команду пропустить, обойти модуль и пройти механообработку на последующих модулях или повторно пройти весь путь и вернуться к этому модулю позже. ЭВМ-супервизор также управляет сменой инструментов в модуле, временем их работы и режимом. Составляются и регулярно печатаются отчеты, выводимые на дисплей, информируя оператора о необходимости замены инструмента. Если емкость магазина позволяет, ЭВМ разрешает использовать инструмент для заранее установленного числа циклов, а затем заменяет его запасным. Кроме того, ЭВМ-супервизор может контролировать и выдавать информацию о выпуске продукции, использовании станка, продолжительности работы, времени простоя и результатах контроля.

2.2. Автоматизированные линии

В качестве примера рассмотрим несколько автоматизированных линий заготовительного производства, по своим характеристикам приближающихся к ГПС кузнечно-прессового производства. Автоматизированная линия горячей штамповки показана на рис. 2.3 [4]. Линия предназначена для получения крупных изделий из крупнотоннажных слитков, производство которых в настоящее время хорошо освоено промышленностью. Линия содержит печь 1 для нагрева крупнотоннажных слитков; автоматизированный ковочный комплекс, состоящий из ковочного прессы 3 и двух манипуляторов 2 с интегрированной системой управления; пресс-ножницы 4 с выдвигным столом для

разделения поковок на мерные длины; печь 5 для нагрева поковок мерной длины для ковки на радиально-ковочной машине 6 с манипуляторами 2; печь 7 для подогрева длинномерных поковок перед разделением их на заготовки мерной длины на пресс-ножницах 8; весы 9 для взвешивания заготовок; кольцевую печь 10 для нагрева заготовок перед предварительной штамповкой на заготовительном прессе 11; печь 12 для нагрева заготовок перед штамповкой на прессе 13 для изотермической штамповки. Линия работает под управлением автоматизированной системы управления 15 и содержит два накопителя 14, которые выполняют роль буферного устройства.

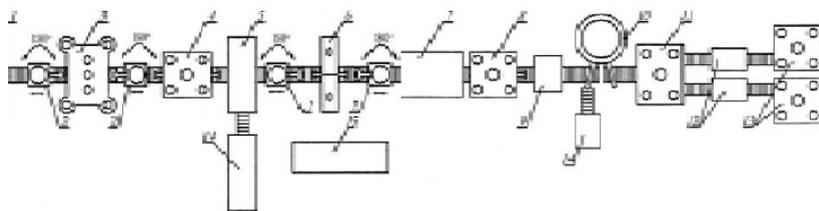


Рис. 2.3. Линия горячей штамповки:

- 1 – печь для нагрева крупных слитков; 2 – манипуляторы;
- 3 – ковочный пресс; 4 – пресс-ножницы для разделения поковок;
- 5 – печь для нагрева поковок; 6 – радиально-ковочная машина;
- 7 – печь для нагрева поковок перед резкой; 8 – пресс-ножницы; 9 – весы;
- 10 – кольцевая печь для нагрева заготовок; 11 – заготовительный пресс для штамповки; 12 – проходная печь; 13 – пресс для изотермической штамповки; 14 – накопитель; 15 – пульт управления

Автоматизированный ковочный комплекс (АКК), входящий в линию, предназначен для ковки крупных слитков на заданный размер. Режимы ковки обеспечивают равномерную «проработку» металла по объему поковки. В соответствии с заказом на пресс-ножницах осуществляют разделение поковки на мерные длины для последующей их ковки на радиально-ковочной машине (РКМ). Режим обжатий на РКМ выбирается из условия получения заданной длины поковки, которая в дальнейшем разделяется на пресс-ножницах на конечное число мерных заготовок

(с учетом удаления концевых остатков). Масса мерных заготовок контролируется на весах, установленных в линии. На заготовительном прессе используется фасонный инструмент для «проработки» центральной части заготовок, которая может быть недостаточно хорошо «проработана» на АКК и РКМ. Далее фасонные заготовки поступают на пресс для изотермической штамповки.

АКК и РКМ оборудованы поворотными манипуляторами, которые могут перемещаться вдоль оси линии. Два накопителя, оборудованные около пресс-ножниц, позволяют в случае необходимости разрывать непрерывный технологический поток и использовать только отдельные участки линии. Пресс-ножницы оборудованы датчиками для измерения размеров поковок, позволяющими получать мерные заготовки одинаковой массы (в случае колебания размеров поковок). Линия сконструирована на базе достаточно хорошо отработанного оборудования для отдельных технологических переделов, прошедшего апробацию в промышленности.

Автоматизированная линия изотермического прессования представлена на рис. 2.4 [4]. Линия содержит печь *I* для нагрева заготовок, на выходе которой установлены датчики *1* температуры; многопозиционное скальпирующее устройство *II* для зачистки поверхности заготовки перед прессованием, снабженное датчиками *2* размеров заготовки и датчиками *3* и *4*, определяющими состояние поверхности заготовки на входе и выходе; универсальный гидравлический пресс *III* для прямого и обратного методов прессования, оборудованный датчиками температуры *5* и *7*, контролирующими температуру прессуемого изделия на выходе из очага матрицы, датчиками *6* скорости движения и положения пресс-штемпеля; электроподогревающее устройство *IV* для дополнительного подогрева прессуемого изделия до температуры закалки, снабженное датчиками *7* и *9* температуры на его входе и выходе; закалочное устройство *V*, обеспечивающее скорость охлаждения, необходимую для закалки изделий из высокопрочных алюминиевых сплавов, снабженное на входе датчиками *8* для определения габаритных размеров прессуемых изделий; правильное устройство *VI* со сканирующим датчиком *10* качества

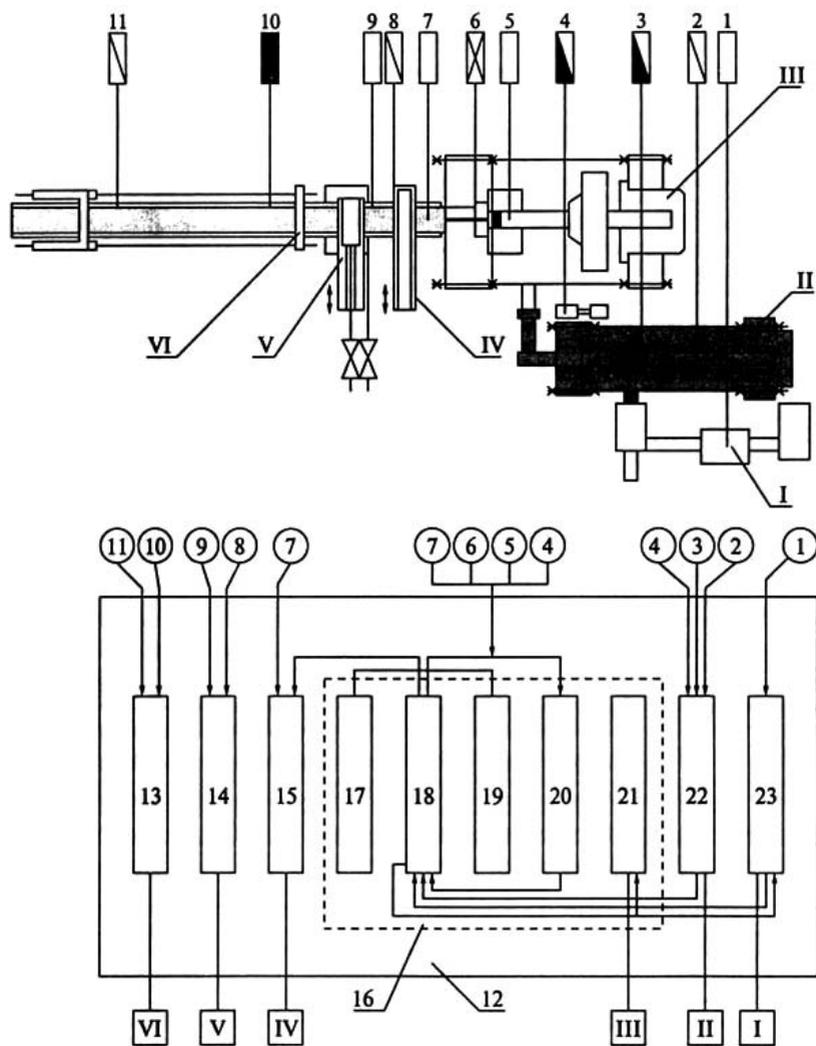


Рис. 2.4. Схема автоматизированной линии изотермического прессования изделий и датчиками II, определяющими размеры элементов прессуемых изделий сложной конфигурации.

Техническая характеристика механического оборудования линии приведена в табл. 2.1.

Таблица 2.1

Техническая характеристика механического оборудования линии

Характеристика	Единица измерения	Величина
<i>Печь для нагрева заготовки</i>		
– температура нагрева	°С	До 500
– время нагрева	с	120
<i>Многопозиционное скальпирующее устройство</i>		
– номинальное усилие	МН	8
– диаметр заготовок	м	0,2...0,5
– длина заготовок	м	0,4...1,0
– скорость рабочего хода главного плунжера	мм/с	до 50
– привод скальпирующего устройства		НАС
– рабочее давление жидкости	МПа	32
– допустимая несоосность заготовок на матрицы при скальпировании	мм	1,0
<i>Универсальный горизонтальный гидравлический пресс для прямого и обратного прессования</i>		
– номинальное усилие	МН	50
– количество ступеней усилия		3
– I ступень	МН	18
– II ступень	МН	32
– III ступень	МН	50
– скорость рабочего хода:		
с плавным регулированием скорости	мм/с	0,1...10,0
со стабилизацией	мм/с	0,1...20,0
– длина контейнера	м	1,2
– внутренний диаметр втулок контейнера	м	0,42
		0,36
		0,31
		0,27
		0,225
высота пресс-остатка:		
– минимальная	м	0,04
– максимальная	м	0,2
давление рабочей жидкости	МПа	32
привод		НАС
<i>Правильно-растяжное устройство</i>		
– номинальное усилие	МН	7
– длина профилей	м	3,0...28,0

Стол прессы оборудован нагревательным и закалочным устройствами, а также устройством для резки изделий (на рис. 2.4 устройство не показано).

Технические показатели линии могут быть изменены в зависимости от применяемого оборудования. Управление технологической линией осуществляется системой управления 12, которая состоит из блока 13 управления правильно-растяжным устройством, блока 14 управления закалочным устройством, блока 15 управления электроподогревательным устройством. Кроме того, система управления содержит блок 16 управления прессом, который, в свою очередь, состоит из блока 17 адаптации, блока 18 моделирования, блока 19 анализатора текущей ситуации, блока 20 управления градиентным нагревом заготовки в контейнере прессы и блока 21 управления исполнительными органами прессы. Управление скальпирующим устройством и температурой нагрева заготовки перед прессованием осуществляется соответственно блоками 22 и 23.

Отличительная особенность системы управления – согласованное двухступенчатое регулирование температуры прессуемого изделия путем управления как скоростью движения пресс-штемпеля, так и дополнительным подогревом прессуемого изделия в электроподогревающем устройстве (перед закалкой), установленном на выходе прессы.

Линия работает следующим образом. Перед началом прессования в автоматизированную систему управления линией вводят основные технологические параметры для прессования конкретного изделия. Затем нагретая в печи I до определенной температуры заготовка поступает на многопозиционное скальпирующее устройство II, где осуществляют зачистку поверхности заготовки (при этом толщина снимаемого слоя металла зависит от размеров и состояния поверхности заготовки). Качество зачищенной поверхности контролирует датчик 4. Далее зачищенная заготовка поступает на пресс III. В процессе прессования результаты измерения температуры поверхности изделия датчиком 7 начинают поступать в блок 19 анализатора текущей ситуации. Исходя из результатов измерений (после их фильтрации и прогнозирования

величины температуры на последующие моменты времени) анализатор текущей ситуации определяет момент, когда продолжение прессования с прежней скоростью может привести к нарушению одной из границ изотермического интервала. В этом случае анализатор устанавливает признак изменения скорости. По этому признаку блок 18 моделирования определяет требуемую скорость прессования на основании текущей оценки параметра, связывающего изменение скорости прессования с изменением температуры выходящего изделия. Вычисленную скорость подают на вход регулятора, поддерживающего заданную величину скорости, воздействуя на исполнительные органы пресса. По окончании теплового переходного процесса, вызванного изменением скорости, блок 17 адаптации корректирует параметр температурно-скоростной зависимости процесса в соответствии с рассогласованием между измеренным и предсказанным значениями температуры. При расчете скорости прессования могут использоваться результаты измерения положения и скорости пресс-штемпеля, что позволяет сделать более гибким алгоритм адаптивного управления. Изделие, выходящее из пресса, поступает в электроподогревающее устройство IV, в котором осуществляют (в случае необходимости) дополнительный подогрев прессуемого изделия до температуры закалки, кроме того, благодаря этому устройству продляется зона высокой температуры металла изделия.

Далее прессуемое изделие поступает в закалочное устройство V. Для предотвращения деформации закалку изделия осуществляют в процессе прессования под нагрузкой.

По окончании прессования и закалки задний конец прессуемого изделия закрепляют в неподвижной траверсе правильно-растяжного устройства VI и осуществляют правку изделия растяжением. Процесс правки (величина растяжения, скоростной, режим правки) контролирует блок 13 управления правильным устройством.

Таким образом, в непрерывном потоке на одном автоматизированном технологическом комплексе осуществляют все основные технологические операции по производству изделий

сложной конфигурации из высокопрочных алюминиевых сплавов, а после небольших изменений и для производства изделий из малолегированных алюминиевых сплавов и тяжелых цветных металлов.

На базе линии изотермического прессования ОАО «Уралмаш» разработан автоматизированный технологический комплекс на базе универсального прессы усилием 50 МН, предназначенный для производства сплошных профилей периодического и постоянного по длине сечений с диаметром описанной окружности до 350 мм.

В состав комплекса входят устройство для зачистки поверхности литых заготовок (скальпирующее устройство) с комплексом автоматики и механизации, горизонтальный гидравлический пресс усилием 50 МН, приемный стол прессы с приводом и устройством для вытягивания отпрессованного изделия из мундштука прессы, устройство для резки изделий на столе прессы, перекладчик отпрессованной продукции с приемного стола прессы на транспортер-холодильник, устройство съема и накопления изделий, автоматизированная система управления температурно-скоростными режимами прессования.

Система управления обеспечивает следующие режимы прессования: изотермическое прессование профилей постоянного сечения; программное изменение скорости прессования для профилей периодического сечения; прессование профилей периодического сечения с постоянной скоростью; программное изменение скорости прессования профилей постоянного сечения. Диапазон изменения и регулирования скорости движения пресс-штемпеля от 0,1 до 10 мм/с, погрешность – 5 %.

Приемный стол прессы – роликовый с приводными роликами. Длина прессуемых профилей: максимальная – 16 м, минимальная – 4,5 м. Устройство для резки изделий на приемном столе прессы обеспечивает резку изделий и пучка изделий. Резке подвергается профиль с диаметром описанной окружности до 350 мм и максимальной толщиной 60 мм. Допустимый диаметр описанной окружности пучка профилей – 250 мм, количество профилей в пучке 1...4 шт.

Комплекс обслуживают три человека: один – на пульте управления, один – на входной стороне комплекса (обслуживает работу индукционной печи и скальпирующего устройства, обеспечивает подачу заготовок и уборку отходов скальпирования), один – осуществляет смену инструмента, уборку партий профилей из накопителя.

Автоматизированные комплексы и линии изотермического прессования позволяют ликвидировать ряд трудоемких технологических операций (повторный нагрев изделий под закалку, закалку изделий, ряд транспортных и подготовительных операций), осуществлять закалку длинномерных изделий определенного сортамента без создания дорогостоящих крупногабаритных закалочных устройств, снизить трудоемкость процесса на 9...10 чел·ч на одну тонну годной продукции; сократить на 350...500 кВт·ч расход электроэнергии на одну тонну изделий; повысить выход годной продукции (за счет снижения отходов) на 2...4 %, что, например, для прутково-профильного прессы усилием 50 МН (при производительности 9000 тонн в год) составит около 300 тонн в год; повысить производительность процесса на 50...60 кг годных изделий в час (за счет повышения скорости прессования); сократить производственные площади; сократить на 2...3 человека обслуживающий персонал; повысить равномерность механических свойств и качество поверхности по всей длине прессуемого изделия за счет изотермического режима прессования и предварительной зачистки поверхности заготовок в многопозиционном скальпирующем устройстве; решить проблему адаптации системы управления скоростью прессования к изменяющимся условиям прессования длинномерных изделий сложной конфигурации.

Описанные выше автоматизированная линия и комплекс изотермического прессования защищены пятью авторскими свидетельствами и патентами ФРГ, Японии, Италии, Турции и Индии.

Схема линии по производству крупногабаритных изделий из высокопрочных алюминиевых сплавов, разработанная ОАО «Уралмаш», показана на рис. 2.5. В состав линии входит следующее уникальное оборудование. Мощный горизонтальный

гидравлический многоцилиндровый пресс, предназначенный для прессования прямым и обратным способами крупногабаритных изделий из алюминиевых сплавов.

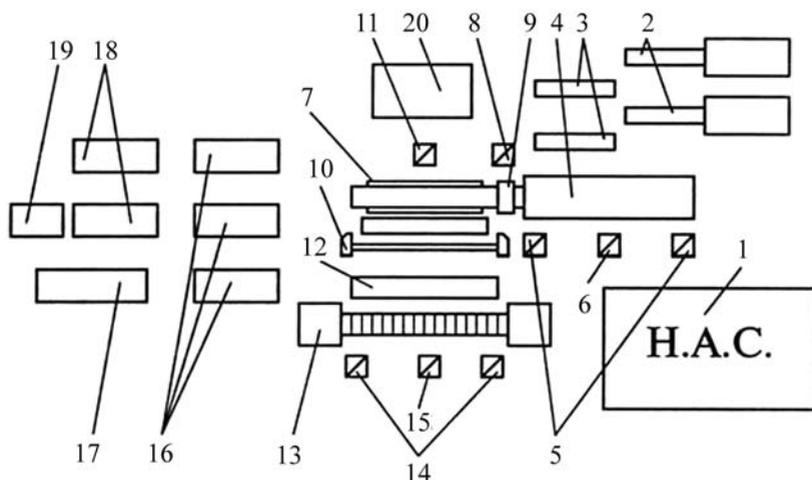


Рис. 2.5. Линия по производству длинномерных крупногабаритных изделий из высокопрочных алюминиевых сплавов:

1 – насосно-аккумуляторная станция; 2 – печи для нагрева слитков; 3 – устройство для зачистки слитков; 4 – горизонтальный гидравлический пресс; 5 – датчики температуры на входе и выходе пресса; 6 – датчик скорости движения пресс-штемпеля; 7 – закалочное устройство на столе пресса; 8 – датчик температуры на входе в закалочное устройство; 9 – подогревающее устройство на столе пресса; 10 – машина для обрезки концов; 11 – датчик качества прессуемых изделий; 12 – стеллаж; 13 – правильно-растяжная машина; 14 – датчик перемещения захватов машины; 15 – датчик для определения деформаций; 16 – горизонтальные агрегаты для закалки крупногабаритных изделий; 17 – роliko-правильная машина; 18 – печи для отжига и старения крупногабаритных изделий; 19 – установки ультразвукового контроля; 20 – центральный пульт автоматизированной системы управления линией прессования

Технические возможности пресса позволяют прессовать изделия с законцовками. Контейнер выполнен удлиненным с повышенным отношением длины контейнера к диаметру рабочей

втулки. Пресс оборудован устройствами для зачистки слитков перед прессованием, стоящими в линии подачи слитков от печи к прессу. Привод пресса, помимо насосно-аккумуляторной станции, оснащен также и индивидуальной насосной установкой с насосами переменной производительности. Схемой предусмотрено подключение любой группы рабочих цилиндров как к насосам, так и к насосно-аккумуляторной станции. На входе и выходе пресса установлены пирометрические бесконтактные датчики температуры для регистрации температуры слитков и прессуемого изделия на выходе из очка матрицы. Скорость движения и положение пресс-штемпеля замеряются с помощью дифференциально-индуктивных датчиков. По своим эксплуатационным характеристикам эти датчики превосходят известные фотоэлектрические датчики, применяемые в локальных регуляторах скорости движения пресс-штемпеля.

Предусмотрена возможность регулировки скорости рабочего хода пресс-штемпеля управляемыми дросселями или насосами переменной производительности. В выходной части пресса имеются приемный стол специальной конструкции, холодильник и пилы для обрезки концов отпрессованных изделий. Все операции от выдачи слитка из печи до получения изделия механизированы.

Правильно-растяжная машина (ПРМ) для правки и упрочнения длинномерных изделий из алюминиевых сплавов состоит из двух тянущих выкрутных головок одинакового исполнения, каждая из которых снабжена индивидуальным масляным приводом. Основанием машины служат станина и опорные балки, установленные и закрепленные на фундаменте. Боковые балки служат опорой и направляющими для проставок. Машина снабжена датчиками скорости и положения тянущих головок, а также сканирующим датчиком деформации отдельных частей пресс-изделия.

Далее в линии установлены горизонтальные агрегаты для закатки крупногабаритных изделий, печи для отжига и старения крупногабаритных изделий, участок ультразвукового контроля и роликоправильная машина.

Технологический процесс производства длинномерных изделий выглядит следующим образом. Слиток, нагретый в печи 2 до температуры, превышающей на 10...15 °С температуру прессования, после зачистки поверхности на устройстве 3 поступает на пресс 4. На прессе обеспечиваются изотермические условия прессования, после чего часть изделий закаляется непосредственно на столе прессы в закалочном устройстве 7. После закалки эта часть изделий, после обрезки концов на машине 10, поступает на ПРМ 13. Контроль данной части изделий производится в потоке. Основная же часть изделий после прессования и обрезки концов поступает на термообработку в агрегатах 16 и 18, а затем на правку на роликовой 17 и протяжной 13 машинах. После этого изделия проходят ультразвуковой контроль на установках 19.

Работа линии координируется автоматизированной системой управления, охватывающей основные технологические операции, такие как нагрев слитков перед прессованием, процесс прессования на гидравлическом прессе, термообработку и правку изделий на растяжной правильной машине. Для части изделий предусмотрен контроль их качества (точность размеров, сплошность, механические свойства) после каждой технологической операции.

Температурный режим нагрева слитков перед прессованием назначается в зависимости от размеров слитков, их химического состава и фактических механических свойств, определяемых в процессе прессования первого слитка. Система управления осуществляет контроль над режимом нагрева.

Известно, что однородность механических свойств по всей длине длинномерных прессуемых изделий имеет место в условиях изотермического прессования, когда обеспечивается постоянство температуры металла на выходе из очка матрицы по всей длине прессуемого изделия. Одновременно достигается и максимальная производительность процесса, что весьма важно при производстве длинномерных изделий.

Система управления процессом прессования призвана обеспечить изотермические условия прессования за счет регулирования скорости движения пресс-штемпеля в процессе прессования.

2.3. Гибкие производственные системы механообрабатывающего производства

Анализ производства большинства промышленно развитых стран показал, что 80...90 % всех деталей, изготавливаемых в настоящее время на станках, целесообразнее получать используя гибкие производственные системы (ГПС или входящие в состав ГПС гибкие производственные модули). Следует иметь в виду, что стоимость программного и аппаратного обеспечения производственного оборудования всей ГПС составляет около 75 % ее стоимости, остальные 25 % стоимости ГПС приходятся на транспортное и другое вспомогательное оборудование. В качестве примера рассмотрим ГПС фирмы Vought (США), предназначенную для механической обработки деталей фюзеляжа американского бомбардировщика В1В. ГПС, поставленная фирмой Cincinnati Milacron (США), включает в себя восемь идентичных горизонтальных обрабатывающих центров с ЧПУ, связанных при помощи транспортной сети из четырех автоматических транспортных тележек с двумя управляемыми от ЭВМ координатно-измерительными машинами; автоматическую станцию мойки и сушки; два карусельных накопителя плит-спутников и автоматическую систему сбора стружки (рис. 2.6).

Станочное оборудование ГПС состоит из восьми горизонтальных обрабатывающих центров HC-20 фирмы Cincinnati Milacron (США), предназначенных для контурного фрезерования, сверления, растачивания, развертывания и нарезания резьбы. Они имеют магазин емкостью 30 инструментов, в котором хранится набор инструментов для обработки на любом из восьми станков. Каждый станок имеет устройство автоматической смены плит-спутников челночного типа с позицией промежуточного хранения. На плитах-спутниках имеются многоместные приспособления для установки различных деталей на одну плиту.

Четырехкоординатные станки могут обрабатывать заготовки размерами 800 × 800 × 900 мм и массой до 2270 кг из черных и цветных металлов. Имеется автоматический конвейер для

отвода стружки, измерительное устройство для установки нулей отсчета, устройство обнаружения поломки инструмента и система адаптивного управления крутящим моментом на шпинделе. Обрабатывающие центры оснащены системами ЧПУ типа CNC Acramatic 900, каждая из которых связана с центральной ЭВМ (DNC).

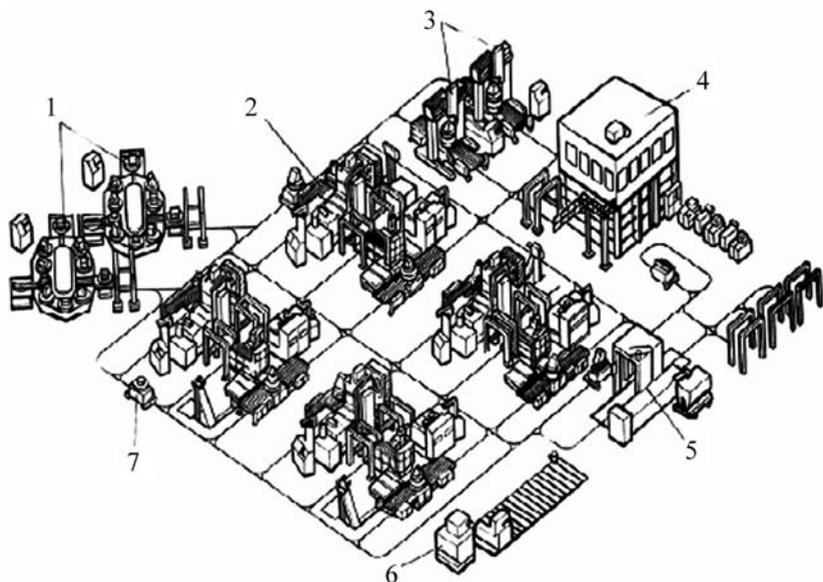


Рис. 2.6. Планировка гибкой производственной системы фирмы Vought: 1 – позиция загрузки-выгрузки; 2 – обрабатывающие центры; 3 – координатно-измерительные машины; 4 – комната с управляющей ЭВМ; 5 – автоматическая моечная станция; 6 – автоматическая система сбора стружки; 7 – направляемые по проводам тележки

Две координатно-измерительные машины производят автоматический контроль обработанных деталей. Данные цикла измерений детали передаются по системе обратной связи на соответствующий станок для автоматической регулировки смещения инструментов. Измерительные машины связаны с системой DNC, циклом измерения управляет центральная ЭВМ. Работу ГПС и других безлюдных систем затрудняет удаление стружки.

Фирма Vought (США) пытается решить эту проблему с помощью автоматической системы сбора стружки и СОЖ, расположенной под полом. Двухпоточная система связана со всеми станками, разделяет стружку черных и цветных металлов, удаляет сгустки масла из СОЖ. Система имеет автоматический самоочищающийся всасывающий фильтр (т. е. в нем не используется сменная фильтровальная бумага), оборудована воздушной системой сушки с электрическим подогревом и конвейером для доставки стружки черных и цветных металлов в различные контейнеры, которые затем транспортируются на автоматических транспортных тележках. Химический состав СОЖ контролируется. Детали и заготовки загружаются и снимаются вручную с двух десятиместных карусельных накопителей, которые в пределах ГПС обслуживаются четырьмя направляемыми по проводам транспортными тележками.

Все ГПС управляет система DNC с помощью ЭВМ. Последовательность обработки деталей в ГПС задается оператором в соответствии с ежедневным производственным заданием. Его решения учитываются в производственном плане предприятия с помощью системы обратной связи.

Фирма Grumman (США) использует интегрированную ячейку для фрезерования и сверления отверстий в деталях сложного профиля и листовом алюминии, которые применяются в конструкции обшивок, панелей, обтекателей и дверей. До внедрения ячейки эти детали обрабатывались вручную с использованием зажимных приспособлений. Центральный элемент ячейки – робот IRE-60 фирмы ASEA. Ячейка включает в себя также пять основных элементов с программным управлением: контейнер, рабочий стапель для установки и закрепления детали, систему автоматической смены инструмента и контроля (рис. 2.7).

В заключение отметим, что в механообработке металлов достигнуты значительные успехи в разработке новых режущих материалов и приводов для металлорежущих станков, в результате чего значительно сократилось основное машинное время на обработку каждой заготовки.

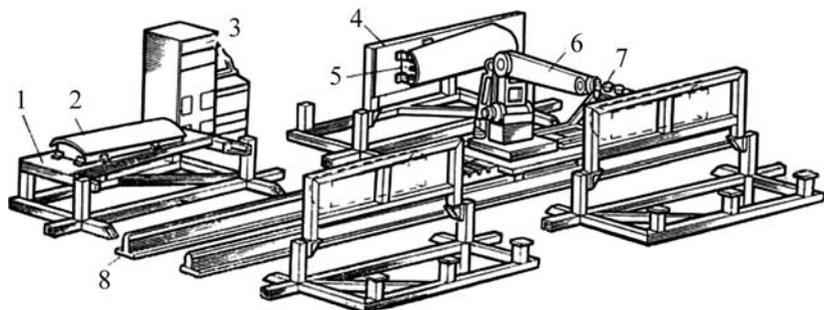


Рис. 2.7. Схема интегральной роботизированной ячейки:
 1 – позиция загрузки; 2 – обрабатываемая деталь; 3 – блок управления;
 4 – позиция обработки; 5 – оснастка; 6 – робот; 7 – робот для смены инструмента; 8 – рельсы для перемещения робота

Много нерешенных проблем существует в области разработки систем крепления, включая системы переналадки приспособлений для крепления инструментов и др. Ведутся дальнейшие работы в области сенсорной техники, систем автоматической регулировки и управления процессами производства. Особое внимание уделяется созданию универсальных и надежных систем распознавания дефектов износа и поломки инструментов. Для большинства металлообрабатывающих предприятий основой развития гибкого производства и снижения затрат на обработку является модульная технология.

Контрольные вопросы

1. Укажите основные виды промышленного производства.
2. Дайте определение и описание ГПС.
3. Дайте определение и описание ГАП.
4. Дайте определение АЛ.
5. Проведите сравнительный анализ ГПС, ГАП и АЛ для различных видов производств.
6. Опишите назначение и структуру автоматизированной линии горячей штамповки.

7. Опишите назначение и структуру автоматизированной линии изотермического прессования.
8. Сформулируйте назначение прессы изотермической штамповки в составе автоматизированной линии горячей штамповки.
9. Сформулируйте принципы изотермического прессования.
10. Сформулируйте принципы системы управления изотермическим прессованием.
11. Опишите функциональное назначение устройств управления линии изотермического прессования.
12. Опишите работу линии изотермического прессования.
13. Укажите технико-экономические характеристики изотермического прессования.
14. Опишите работу линии по производству длинномерных крупногабаритных изделий из высокопрочных алюминиевых сплавов.
15. Опишите планировку гибкой производственной системы Vought.
16. Дайте технико-экономическую характеристику гибкой производственной системы Vought.

Лекция 3

ВВЕДЕНИЕ В ПРОЦЕСС ПРОЕКТИРОВАНИЯ МЕХАТРОННЫХ СИСТЕМ

Проектирование технического объекта – создание, преобразование и представление в принятой форме образца этого, еще не существующего, объекта. Образ объекта может создаваться в воображении человека в результате творческого процесса или генерироваться с помощью некоторых алгоритмов в процессе взаимодействия с ЭВМ.

Инженерное проектирование начинается при наличии выраженной потребности в некоторых технических объектах. Результатом проектирования, как правило, служит полный комплект документации, содержащий сведения, достаточные для изготовления объекта в заданных условиях. Эта документация и есть проект – окончательное описание объекта.

Проектирование предполагает выполнение комплекса работ исследовательского, расчетного и конструктивного характера.

Проектирование, при котором все проектные решения или их часть получают путем взаимодействия человека с ЭВМ, называется *автоматизированным*. Система, реализующая автоматизированное проектирование, представляет собой *систему автоматизированного проектирования* – CAD (Computer Aided Design в англоязычной терминологии).

Имеется несколько типов проектов: технические, организационные, экономические, социальные и смешанные. Однако процесс проектирования имеет две особенности: во-первых, состав и последовательность его этапов не зависит от целевого назначения проекта, во-вторых, логика процесса проектирования инвариантна к способу проектирования – традиционному или автоматизированному.

Кроме того, для подходов к проектированию сложных мехатронных систем характерны:

- структуризация процесса проектирования (декомпозиция проектных задач и документации);
- итерационный характер проектирования;
- типизация и унификация проектных решений и средств проектирования;
- информационная поддержка (в том или ином виде) на всех этапах процесса проектирования;
- соблюдение основных принципов проектирования.

3.1. Основные принципы проектирования

При решении задач проектирования руководствуются рядом основных принципов.

Последовательность и итерационность. Последовательность заключается в строгой очередности выполнения этапов проектирования механизма или машины, а итерационность – в корректировках проектных решений, полученных на предыдущих этапах проектирования.

Модульный принцип проектирования состоит в максимально возможном использовании однотипных узлов (или элементов узлов) при проектировании семейства машин различного целевого назначения. Сложные системы необходимо проектировать так, чтобы поведение одной части системы оказывало минимальное воздействие на остальную систему. Этот принцип базируется на активном использовании компьютерной техники и применяется тогда, когда уже имеется достаточно четкое представление о семействе проектируемых машин.

Применение модульного принципа при проектировании одного механизма или одной машины бессмысленно и невозможно. Только в случае разработки семейства машин и использовании при этом ограниченного числа готовых узлов последние становятся модулями. Модули являются следствием принятого технического решения, ибо в принципе проектировать можно так, что

узлы одного назначения не будут даже отдаленно похожи друг на друга.

Различают модули *производственные* (применяемые без каких-либо изменений в машинах различного целевого назначения), *технологические* (отличающиеся в основном технологией изготовления и незначительными конструкторскими изменениями, относящимися, как правило, к местам крепления узлов, например: правый-левый, верхний-нижний, передний-задний и т. п.) и *конструкционные* (имеющие конструкционное подобие, но различные размеры).

Современное техническое обеспечение САПР позволяет легко получать зеркальные изображения узлов (технологические модули) с минимальными затратами труда и времени. Несколько сложнее получить конструкционные модули, так как при их проектировании необходимо одновременно выполнять требования соблюдения размерных рядов и максимально возможной унификации.

При использовании модульного принципа проектирования на уровне принятия концепции должен быть решен вопрос о критерии целесообразности его применения для проектируемого семейства машин. Модули, используемые в последующих разработках, переходят в категорию унифицированных узлов.

Принцип унификации связан с применением в семействе проектируемых механизмов или машин унифицированных сборочных единиц (узлов, подузлов, агрегатов), деталей (оригинальных и стандартных), комплектующих. Как показывает опыт автоматизированного проектирования, применение типовых унифицированных деталей нецелесообразно. Значительно выгоднее иметь большое количество макрусов и существенные заделы в банках данных по модификациям и унифицированным узлам, что позволяет избегать излишней детализации конструкторских разработок. Информацию об унифицированных деталях, узлах и системах целесообразно хранить на микрофишах систематизированно по группам.

Принцип соответствия выбора номенклатуры и значений выходных характеристик целевому назначению проектируемой

машины или механизма. Чем ответственнее проектируемый объект, тем большее число выходных характеристик и параметров объекта регламентируется и тем более жесткие требования к ним предъявляются. Например, вероятность безотказной работы станков принимается в пределах 0,95–0,99, а самолетов – 0,999 99.

Другой пример. Число установленных и проверяемых выходных параметров точности станка при высоких требованиях к точности обрабатываемых деталей может достигать 20–30, в то время как для станков нормальной точности достаточно регламентировать 8–10 показателей. Основная цель регламентации выходных параметров станка – обеспечить погрешность обработки, которая находилась бы в установленных пределах в течение всего периода его эксплуатации.

Во всех случаях проектировщику необходимо помнить, что соответствие выходных характеристик механизма или машины их целевому назначению в первую очередь определяет общая компоновка, принимаемая на концептуальном уровне.

Принцип компромиссов. Проектирование – это непрерывная цепь компромиссов, которые приходится принимать на всех стадиях создания механизма или машины. Например, стремление к увеличению частоты вращения шпиндельных узлов всегда приводит к увеличению тепловыделения в опорах, которое не может превышать определенного уровня, и принятие окончательного решения всегда требует компромисса.

Улучшение любой технической характеристики машины (скорости, грузоподъемности, точности, надежности, производительности и др.) неизбежно вызывает увеличение ее стоимости, трудоемкости изготовления, часто требует другого уровня обслуживания и повышения культуры эксплуатации, что всегда приводит к компромиссным решениям.

Тенденция современных машиностроительных производств к концентрации операций постоянно требует от проектировщика решения задачи рационального сочетания технологических возможностей и усложнения конструкций оборудования. Избыточность технологических возможностей может быть не оправдана экономически.

Увеличение надежности машины почти всегда сопровождается ее усложнением и удорожанием. Выбор опор и направляющих, обеспечивающих незначительный износ, применение новых материалов, специальных покрытий, смазок, термообработки повышают трудоемкость изготовления и стоимость машины. Далеко не всегда ясно, *что* экономически целесообразнее – удорожать механизм или машину изначально либо совершенствовать систему обслуживания и ремонта в период эксплуатации. Компромисс заключается в обеспечении в первую очередь надежности наиболее ответственных деталей и узлов машины.

Принцип преемственности. Практически любые механизмы и машины являются продуктом эволюции, и в них всегда имеются элементы, детали и узлы, разработанные ранее. Соблюдение преемственности является одним из эффективных путей снижения затрат и сокращения сроков создания машины.

Хорошо отработанные элементы и узлы нет нужды заменять, и в одной машине прекрасно могут сосуществовать «новое» и «старое». Изыскание новых принципов построения механизмов и машин не является, как правило, задачей конструирования. *Суть проектирования – обеспечение требуемых характеристик машины, а не погоня за оригинальностью.* Преемственность определяется самой логикой последовательного развития машин. Модернизация, систематическое совершенствование позволяют поддерживать показатели выпускаемых машин на уровне постоянно возрастающих требований в течение определенного времени без коренной переработки конструкции. Например, ведущие автомобильные фирмы выпускают принципиально новую модель только раз в 8–10 лет.

Вопрос заключается в том, как *наиболее рационально сочетать новое со старым.* Это зависит как от качества отработки отдельных проектных решений (стоит ли заменить), так и от уровня технологической подготовки производства (можем ли заменить). На этом пути существуют две основные опасности: консерватизм, влекущий за собой добровольный отказ от творческого поиска и приводящий к необъективной критике новых решений и отказу от них, и психологическая инерция, ослабляющая

самоконтроль, в результате чего конструктору очень трудно отделиться от своего решения, так как он находится в плену идеи.

Преимственность при проектировании имеет несколько аспектов:

- приверженность фирм к определенным, отработанным годами решениям (в области компоновки, конструкции узлов, дизайна и т. п.). Это то, что определяет «почерк» фирм, и они весьма неохотно расстаются со своими традиционными решениями, часто связанными с «ноу-хау» и даже иногда носящими названия фирм;

- индивидуализация конструкций механизмов и машин не должна распространяться на все элементы и узлы. Модули могут и должны использоваться для различных моделей;

- конструктивная преимственность дает возможность увеличить эффективность разработки за счет использования модулей и унифицированных элементов и узлов;

- степень заимствования определяется на концептуальном уровне и зависит от возможности принятия своих и чужих технических решений в неизменном или модифицированном виде, реальных возможностей проектировщика, определяемых квалификацией, сроками и другими обстоятельствами.

Работающая сложная система неизбежно оказывается результатом работающей простой системы. Сложная система, разработанная на бумаге от начала и до конца, как правило, никогда не работает, и ее невозможно заставить работать. Всегда надо начинать с работающей простой системы.

3.2. Схема процесса проектирования

Стадии проектирования – наиболее крупные части проектирования процесса, развивающегося во времени. В общем случае выделяются стадии научно-исследовательских работ (НИР), эскизного проекта или опытно-конструкторских работ, технического, рабочего проектов, испытаний опытных образцов или опытных партий. Стадию НИР иногда называют предпроектными

исследованиями, или стадией технического предложения. Очевидно, что по мере перехода от стадии к стадии степень подробности и тщательность проработки проекта возрастают. Рабочий проект должен быть вполне достаточным для изготовления опытных или серийных образцов. Близким к определению стадии, но менее четко оговоренным понятием является понятие этапа проектирования.

Стадии (этапы) проектирования подразделяют на составные части, называемые **проектными процедурами**. Примерами проектных процедур могут служить подготовка детализированных чертежей, анализ кинематики, моделирование переходного процесса, оптимизация параметров и другие проектные задачи.

В свою очередь, проектные процедуры можно расчленить на более мелкие компоненты, называемые **проектными операциями**. Например, при анализе прочности детали сеточными методами могут быть построены сетки, произведен выбор или расчет внешних воздействий, собственно моделирование полей напряжений и деформаций, представление результатов моделирования в графической и текстовой формах. Проектирование сводится к выполнению некоторых последовательностей проектных процедур – **маршрутов проектирования**.

На рис. 3.1 показана укрупненная схема процесса проектирования.

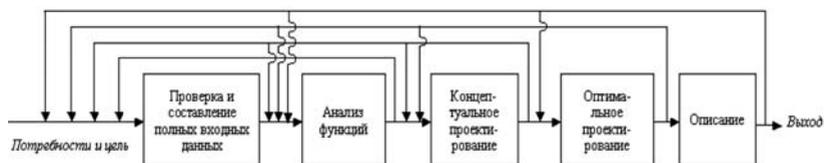


Рис. 3.1. Укрупненная схема процесса проектирования

Техническая разработка системы начинается с определения нужд потенциального пользователя разрабатываемой системы. Часто оказывается, что пользователь желает иметь систему, выполняющую данную работу, однако испытывает трудности при количественной формулировке своих потребностей и целей.

Поэтому у инженера и пользователя возникает дополнительная задача количественного воплощения целей системы так, чтобы была определена содержательная совокупность целевых функционалов для последующей разработки (проверка и составление полных входных данных).

Когда скоро потребности и цели системы определены, необходимо дать количественное описание функций, которые должны выполнять система и любая требуемая подсистема. Эта процедура называется анализом функций. Его цель состоит в выборе функций, или операций, которые должны быть проведены для выполнения задачи, требуемой от разрабатываемой системы. Эти функции становятся затем целями низшего уровня для разработки подсистем. Определение функций по своей природе носит качественный характер. Однако когда скоро функция или операция определена, она должна быть описана в количественных терминах. Например, если операцию нужно провести быстро, необходимо оговорить допустимое время.

Следующий шаг, показанный на рис. 3.1, является одним из наиболее трудных этапов в технической разработке системы и, конечно, самым трудным шагом для аналитического описания. Концептуальное проектирование, как видно из названия, состоит в определении концепций, или основных конфигураций системы, которые могут удовлетворить целям системы. На этом шаге желательно оставить концепции по возможности в общем виде, с тем чтобы не исключить кандидатуры систем, которые могут оказаться эффективными. Например, если операция, подлежащая выполнению, есть продвижение летательного аппарата над поверхностью Земли, то концептуальные проекты могут включать в себя проектирование колес, направляющих, стоек и воздушной подушки. На этом этапе процесса проектирования важно определить границы приемлемых значений параметров, описывающих систему. Для параметров проектирования в пределах этих значений система должна уметь выполнять функции, определенные на предыдущем шаге.

Пусть мы имеем дело с оптимальным проектом, для которого целью является выбор неопределенных параметров, введенных

на предыдущем шаге (оптимальное проектирование). Эти параметры должны быть в пределах, определяемых технологическими ограничениями и назначением системы. Критерием выбора параметров системы является максимизация стоимости системы или минимизация меры затрат. Следует особо отметить, что математически точный оптимум может оказаться недостижимым и, следовательно, служить лишь ориентиром. Однако методы выбора параметров системы должны обладать таким свойством, что если оптимум существует, то при достаточном терпении и заданном машинном времени он должен быть в пределе достижим.

То что является последним шагом в модели технической разработки системы на рис. 3.1 (описание оптимальной системы), в действительности оказывается промежуточным шагом. До тех пор пока процедура проектирования системы не станет исключительно эффективной, система, предложенная группой разработчиков, не будет удовлетворять заказчика. Получив результаты одного прохождения через процедуру проектирования системы, заказчик, вероятно, вспомнит некоторые ограничения, которые он забыл описать и которые оптимальная система нарушает. Проектировщик также может найти привлекательные решения, которых он ранее не видел. Он опять-таки в большей мере, чем заказчик, вспомнит технологические ограничения, которые он забыл описать и которые оптимальная система нарушает. Наконец, заказчик, несомненно, решит, что неплохо бы уменьшить на малую величину уровни качества системы, если это приведет к экономии денег.

На следующем шаге процедуры каждый участник группы должен сделать глубокий вдох и вернуться к работе, вооруженный добытыми с трудом новыми знаниями. Именно с этой целью указаны все каналы обратной связи в модели на рис. 3.1. Эта итеративная процедура продолжается до тех пор, пока заказчик не решит, что предлагаемая система является той, в которой он действительно нуждается. Это будет еще одно решение, принимаемое человеком, но математически не запрограммированное.

Основные стадии и процедуры процесса проектирования показаны на рис. 3.2. Разделение процесса проектирования на последовательные стадии (этапы) является в известной степени

условным, поскольку в процессе проектирования пересматриваются и уточняются ранее принятые решения.

Прокомментируем основные стадии проектирования. **Техническое задание** (ТЗ) обосновывает те новые качества, которыми должен обладать проектируемый объект, и является наряду с техническим предложением итогом предпроектной подготовки. ТЗ устанавливает основное назначение объекта, обосновывает целесообразность его создания и регламентирует все основные технические характеристики (требования).

В процессе подготовки ТЗ разрабатывается основная концепция проектирования. Разработкой концепции занимается небольшая по численности группа проектантов высшей квалификации.

Техническое предложение формулирует принятую концепцию, уточняя и развивая техническое задание. Оно состоит из совокупности конструкторских документов, необходимых для дальнейшего проектирования. Техническое предложение обеспечивает и уточняет технические характеристики проектируемой машины и диапазоны эксплуатационных нагрузок и скоростей. На этом этапе выбирают окончательный вариант компоновки машины, ее узлов и агрегатов, пользуясь результатами анализа, синтеза и оптимизации вариантов компоновок, и разрабатывают принципиальные схемы машины: кинематическую, гидравлическую, пневматическую, электрическую и др., а также принимают тип и характеристики системы управления, определяют общие габаритные размеры.

Техническое предложение дает полное представление о компоновке и возможностях проектируемых механизмов или машин.

Эскизный проект содержит предварительную конструкторскую проработку всех основных узлов и является развитием технического предложения. Он базируется на анализе различных конструкторских решений, результатах расчетов, оптимизации важнейших параметров и характеристик машины.

При конструировании машины, ее узлов и систем следует максимально использовать стандартные и унифицированные детали, механизмы и элементы, что удешевляет проектируемую машину.

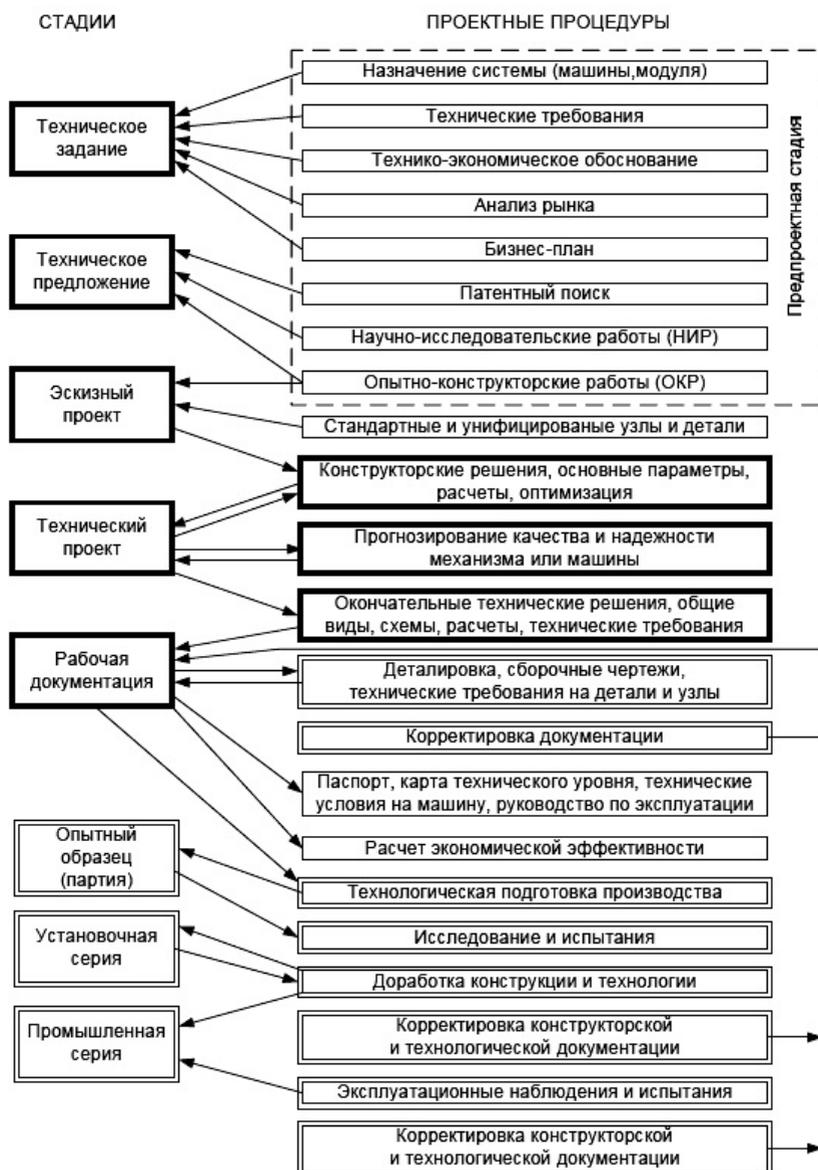


Рис. 3.2. Стадии и процедуры процесса проектирования

Каждое конструкторское решение должно быть обосновано результатами расчетов, рассмотрено с точки зрения обеспечения качества и надежности функционирования проектируемой машины, технологичности изготовления и сборки, удобства эксплуатации, ремонтпригодности и т. д.

Технический проект включает в себя окончательную конструкторскую проработку всех узлов, чертежей, схем машины и ее общих видов. На этом этапе производят все виды уточненных и поверочных расчетов, оптимизируют необходимые параметры узлов и систем машины, окончательно определяют ее эффективность; прогнозируют выходные характеристики и оценивают надежность машины с учетом вероятностной природы режимов ее эксплуатации. Проект должен содержать все исходные данные, необходимые для разработки комплекта технической документации на проектируемый механизм или машину, его согласовывают и утверждают в установленном порядке.

Рабочая документация – завершающий этап проектирования. Он включает в себя разработку рабочих чертежей всех оригинальных деталей и простановку технических требований на их изготовление, составление технологической документации, корректировки (в случае необходимости) технического проекта, составление спецификаций оригинальных и покупных комплектующих деталей, формирование паспорта машины, карты технического уровня, инструкции по эксплуатации и ряда других документов. Рабочая документация должна содержать все данные, необходимые для изготовления проектируемого механизма или машины. Выпуск рабочей документации существенно упрощается при применении САПР.

После завершения разработки комплекта рабочей документации осуществляются **технологическая подготовка производства** и последующие этапы изготовления и отладки спроектированной машины. В ряде случаев для сокращения сроков выпуска новой техники технологическую подготовку производства начинают заранее до завершения выпуска комплекта рабочей документации. Применение техники CAD/CAM значительно сокращает сроки подготовки производства.

3.3. Системный подход к проектированию

В лекции 1 было отмечено, что наиболее общим подходом к проектированию мехатронных систем является системный подход. В системотехнике можно выделить три подхода к процессу проектирования: структурный, блочно-иерархический и объектно-ориентированный.

Блочнo-иерархический подход к проектированию использует идеи декомпозиции описания сложных систем на иерархические уровни, вводит понятие стиля проектирования, устанавливает связь между параметрами соседних иерархических уровней.

В лекции 1 отмечено, что для большинства приложений характерны три иерархических уровня: *системный*, *макроуровень* и *микроуровень*. В каждом – приложение наименований уровней и их число может быть различным (в радиотехнике, вычислительной технике, машиностроении и т. д.). Например, в машиностроении имеются уровни: деталей, узлов, машин и комплексов машин.

В зависимости от последовательности решения задач иерархических уровней различают *нисходящее*, *восходящее* и *смешанное* проектирование (стили проектирования).

Восходящее проектирование предполагает решение задач от нижних уровней к верхним. *Нисходящее* проектирование – от верхних к нижним. *Смешанный стиль* имеет элементы как восходящего, так и нисходящего проектирования. В большинстве случаев для сложных систем предпочтение отдают нисходящему проектированию, но часто используют и смешанное.

Неопределенность и нечеткость исходных данных при нисходящем проектировании (так как еще не спроектированы компоненты) или исходных требований при восходящем проектировании (поскольку техническое задание имеется на всю систему, а не на ее части) обуславливают необходимость прогнозирования недостающих данных с последующим их уточнением, т. е. последовательного приближения к окончательному решению (итерационность проектирования).

Наряду с декомпозицией описаний на иерархические уровни применяют разделение представлений о проектируемых объектах на аспекты.

Аспект описания (страта) – описание системы или ее части с некоторой оговоренной точки зрения, определяемой функциональными, физическими или иного типа отношениями между свойствами и элементами.

Различают **функциональный, информационный, структурный и поведенческий** (процессный) **аспекты**.

Функциональное описание относят к функциям системы и чаще всего представляют его функциональными схемами.

Информационное описание включает в себя основные понятия предметной области (сущности), словесное пояснение или числовые значения характеристик (атрибутов) используемых объектов, а также описание связей между этими понятиями и характеристиками. Информационные модели можно представлять графически (графы, диаграммы *сущность – отношение*), в виде таблиц или списков.

Структурное описание относится к морфологии системы, характеризует составные части системы и их межсоединения и может быть представлено структурными схемами, а также различного рода конструкторской документацией.

Поведенческое описание характеризует процессы функционирования (алгоритмы) системы и (или) технологические процессы создания системы.

Иногда аспекты описаний связывают с подсистемами, функционирование которых основано на различных физических процессах.

Отметим, что в общем случае выделение страт может быть неоднозначным. Так, помимо указанного подхода очевидна целесообразность выделения таких аспектов проектирования, как *функциональное* (разработка принципов действия, структурных, функциональных, принципиальных схем), *конструкторское* (определение форм и пространственного расположения компонентов изделий), *алгоритмическое* (разработка алгоритмов и программного обеспечения) и *технологическое* (разработка технологических процессов) проектирование систем. Примерами страт в случае САПР могут служить также рассматриваемые далее виды обеспечения автоматизированного проектирования.

Структурный подход к проектированию требует синтезацию вариантов системы из компонентов (блоков, модулей) и оценку этих вариантов (предварительно спрогнозировав характеристики компонентов). Он основан на идее алгоритмической декомпозиции (по методу «сверху-вниз»), где каждый блок (модуль) системы выполняет один из этапов общего процесса проектирования. Таким образом, представление о проектируемой системе расчленяется на отдельные блоки (модули), являющиеся частями алгоритма функционирования системы.

Структурный подход включает в себя создание структуры проектируемой системы и расчет ее параметров. Эти две части подхода называют *структурным* и *параметрическим синтезом* системы (как это было сказано в лекции 1). Задачи структурного синтеза относятся к задачам проектирования, наиболее сложным с точки зрения возможности их формализации.

Результатом структурного синтеза должны быть или перечень типов элементов вместе с таблицей соединений, или схема расположения элементов с указанием их типа, или эскизный чертеж, или схема алгоритма и т.п. Структурный синтез обычно осуществляется по блочно-иерархическому принципу. Это означает, что синтезируется не вся сложная система целиком, а на каждом уровне проектирования синтезируется определенный уровень системы – структурная схема (элементы изображения устройств), далее функциональная схема и конструкции устройств (элементы, блоки), затем более мелкие функциональные и конструктивные единицы. Такой подход существенно упрощает решение задачи синтеза.

Параметрический синтез систем обычно реализуют с помощью методов и алгоритмов оптимизации. Однако для непрерывных систем (когда управляемые параметры непрерывны и могут принимать любые значения в пределах непрерывной допустимой области) и для дискретных систем (с дискретными внутренними параметрами) методы оптимизации имеют существенное отличие. Задачи дискретной оптимизации относятся к задачам дискретного математического программирования.

Задачи дискретного программирования считаются более трудными для решения, чем задачи математического программирования с непрерывными параметрами.

Сложность решения задач синтеза систем естественно отождествлять со сложностью производственных систем. Для поддержки концептуального проектирования сложных плохо структурированных систем наиболее перспективными являются генетические методы оптимизации, экспертные системы и CASE-системы, которые будут рассмотрены в последующих лекциях.

Объектно-ориентированный подход рассматривает сложную систему как совокупность взаимодействующих друг с другом объектов, каждый из которых является экземпляром отдельного класса. Такой подход наиболее перспективен при проектировании и моделировании сложных систем.

При структурном подходе жизненный цикл разработки сложной системы складывается из этапов анализа, проектирования, программирования, тестирования и сопровождения, которые выполняются последовательно. Главным недостатком такого подхода является последовательность выполнения этапов. Например, программирование можно начать только по завершении анализа и проектирования. Это приводит к большим потерям времени, не позволяет быстро разрабатывать прототипы программной системы. Этот подход не согласуется с интерактивным характером разработки программной системы, поскольку на последних этапах может выясниться необходимость внесения изменений в решения, принятые на предыдущих этапах (см. лекцию 1).

В целом структурному подходу присущи следующие недостатки [6]:

- трудоемкость внесения изменений;
- большой объем документации по проекту;
- серьезные ограничения возможностей сборки системы из готовых компонентов;
- сложность переноса на другие платформы.

Стремление избавиться от недостатков структурного подхода, базирующегося на системном анализе, привело к возникновению объектно-ориентированного анализа (ООА). В ООА такие понятия, как система, структура, состояние, иерархия, событие,

сущность, пришедшие из системотехники, были дополнены новыми понятиями, такими как *домен, класс, объект, атрибут, наследственность, инкапсуляция, полиморфизм* и др.

Долгое время объектно-ориентированный подход (ООП) (на базе ООА) развивался почти исключительно программистами, хотя в начале он предназначался для моделирования (проектирования) сложных систем и использовался в языке моделирования Simula-67. Своеобразным итогом многолетнего развития объектно-ориентированного программирования можно считать появление «унифицированного языка моделирования» UML (Unified Modeling Language).

Только в последние годы ООП стал востребованным в своей первоначальной области – моделировании. Для его реализации наиболее известны пакеты Simulink и Stateflow, работающие вместе с Matlab.

Рассмотрим несколько основополагающих понятий объектно-ориентированного анализа (ООА): *сущность, связь, атрибут*.

Под *сущностью* (entity) понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может являться экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности. Примерами сущностей могут быть: предприятие, фирма, цех, банк, клиент банка, технология, компьютер, рейс. Каждая из сущностей может рассматриваться с различной степенью детализации, что определяется конкретной постановкой задачи.

Связь (relationship) определяется как отношение или некоторая ассоциация между отдельными сущностями. Примером связей могут быть родственные отношения типа «отец – сын» или производственные отношения «начальник – подчиненный». Другой тип связей задается отношениями «иметь в собственности», «обладать свойством» или «быть частью».

Атрибут – это каждая отдельная характеристика, которая является общей для всех возможных экземпляров объекта.

Атрибутами являются такие характеристики, как высота, температура, регистрационный номер или положение предметов в реальном мире.

Чтобы обратиться к атрибуту, пишут <имя объекта>, <имя атрибута>, как например:

- самолет, размах крыла;
- листопрокатный стан, 1500;
- аудитория, лекционная.

Фундаментальными понятиями объектно-ориентированного подхода являются понятия *объекта*, *класса*, *домена*. *Объекты* – конкретные предметы, а также абстрактные и реальные сущности в реальном мире, а точнее, в интересующей разработчика предметной области. Например, объектами могут быть: студент, кафедра, университет. Объект обладает индивидуальностью и поведением, имеет атрибуты, значение которых определяют его состояние. Так, конкретный студент, сдавая экзамен, может оказаться в состоянии, когда его ответы не удовлетворяют экзаменатора, а его «поведение» в этом случае заключается в «дополнительной подготовке к экзамену».

Класс – это абстрактная совокупность объектов, которые имеют общий набор свойств и обладают одинаковым поведением. Понятие класса впервые было введено в язык UML. Объекты и классы обладают характерными свойствами, которые активно используются при объектно-ориентированном подходе и во многом определяют его преимущество. К важной особенности классов относится возможность их организации в виде некоторой иерархической структуры.

Домен – отдельная предметная область, «населенная» набором объектов, которые ведут себя в соответствии с правилами домена. Большие домены разбивают на подсистемы.

Основными принципами объектно-ориентированного подхода являются: *наследование*, *инкапсуляция* и *полиморфизм*.

Наследование – первый принцип, в соответствии с которым знание о более общей категории разрешается применять для более частной категории. Наследование тесно связано с иерархией классов, которая определяет, какие классы следует считать

наиболее абстрактными и общими по отношению к другим классам. При этом если родительский класс (предок) обладает фиксированным набором свойств и поведением, то производный от него класс (потомок) должен содержать этот же набор свойств и поведение, а также дополнительные свойства, которые будут характеризовать уникальность полученного таким образом класса. В качестве примера можно привести общий класс «Автомобиль», а в качестве производных классов рассмотреть «Легковой автомобиль», «Легковой автомобиль производства BMW», «Конкретную модель BMW». Наследование, позволяющее создавать иерархию классов, является эффективным средством внесения изменений и дополнений в программы проектируемых систем.

Инкапсуляция – это сокрытие отдельных деталей внутреннего устройства классов от внешних по отношению к нему объектов или пользователей. Объекты можно считать самостоятельными сущностями, отделенными от внешнего мира. Для того чтобы объект произвел некоторое действие, ему извне необходимо послать сообщение, которое инициирует выполнение нужного действия. Инкапсуляция позволяет изменять реализацию любого класса объектов без опасения, что это вызовет нежелательные побочные эффекты в программной системе. Тем самым упрощается процесс исправления ошибок и модификацию программ.

Полиморфизм – это свойство некоторых объектов принимать различные внешние формы в зависимости от обстоятельств. Применительно к объектно-ориентированному программированию (ООП) полиморфизм означает, что действия, выполняемые одноименными методами, могут отличаться в зависимости от того, к какому из классов относится тот или иной метод [7]. Рассмотрим, например, три объекта соответствующих классов: «Студент», «Водитель», «Фигурист». Для каждого из них можно определить операцию учиться. Однако результат выполнения этой операции будет отличаться для каждого из рассмотренных объектов.

Методология ООП оказала влияние на процесс разработки программ. *Процедурно-ориентированная декомпозиция* программ уступила место *объектно-ориентированной декомпозиции*, при которой отдельными структурными единицами программы стали

являться не процедуры и функции, а классы и объекты с соответствующими свойствами и методами (действиями). Программа стала *событийно-управляемой* и перестала быть последовательностью predetermined на этапе кодирования действий. При разработке современных приложений каждая программа представляет собой бесконечный цикл ожидания некоторых заранее определенных событий. При этом инициаторами событий могут быть другие программы или пользователи.

При использовании методологии ООП написание программного кода может быть отделено от процесса проектирования структуры программы. До начала программирования классов, их свойств и методов необходимо определить, чем же являются сами эти классы. Более того, нужно дать ответы на такие вопросы, как: сколько и каких классов необходимо определить для решения поставленной задачи, какие свойства и методы необходимы для придания классам необходимого поведения, а также установить взаимодействие между классами. Таким образом, анализ предметной области решаемой задачи необходимо проводить до начала написания программы.

Жизненный цикл объектно-ориентированной разработки программных систем содержит несколько этапов, но в отличие от структурного подхода в нем нет строгой последовательности их выполнения. Процесс носит принципиально итерационный характер, что значительно упрощает написание программ [6].

Разработка начинается с этапа исследования (объектно-ориентированного анализа). Здесь предъявляются требования к системе. Затем осуществляется анализ предметной области, в ходе которого определяются классы и объекты, которые составляют словарь предметной области.

После исследования начинается объектно-ориентированное проектирование, в ходе которого детализируются классы и объекты, полученные на этапе анализа. При этом могут вводиться новые классы и объекты, если это потребует. В результате проектирования должна быть создана детальная модель системы, составлены спецификации объектов, классов и отношений, достаточных для программирования.

Для разработки программных систем используются инструментальные средства CASE-систем, такие как SADT – Structured Analysis and Design Technique, IDEF – Integrated DEFinition и UML. Эти средства будут рассмотрены в последующих лекциях.

В отличие от процедурно-ориентированного подхода к программированию, в объектно-ориентированном подходе к моделированию и проектированию реальных физических систем различного назначения полиморфизмом называется возможность использования вместо объектов одного декларированного класса объектов другого класса, называемого замещающим, совместно с первым. Совместимость классов означает, что либо замещающий класс является потомком декларированного класса, либо декларирован интерфейс, а замещающий класс реализует этот интерфейс. Таким образом, можно выделить «совместимость по наследованию» и «совместимость по интерфейсу». Под интерфейсом в объектно-ориентированном моделировании (ООМ) следует понимать некоторую совокупность видимых компонентов объекта, т.е. для блока (на структурной схеме объекта) – это внешние переменные и видимые извне процедуры и функции. Таким образом, если блок (замещающего класса) содержит совокупность видимых компонентов с теми же именами, что в декларации интерфейса, и с совместимыми типами, то блок реализует этот интерфейс. Каждый блок неявно задает определенный интерфейс – совокупность всех своих внешних компонентов [8].

Особенность процесса разработки современных сложных систем состоит в том, что центр тяжести работ смещается от программирования к более ранним этапам – анализу и проектированию.

Рассмотрим основные преимущества объектно-ориентированного подхода [6]:

- *распараллеливание работ* (программирование и тестирование отдельных компонентов системы возможно до завершения проектирования, возможность внесения изменений и дополнений без переработки проекта);
- *гибкость архитектуры и переносимость* (в клиент-серверной системе объекты могут размещаться как на местах клиента,

так и на сервере. Возможность реализации классов на компьютерах различного типа, а фиксированный интерфейс каждого класса обеспечивает правильность функционирования системы);

– *возможность повторного использования программных компонентов* (библиотеки классов позволяют значительно снизить объем программирования новых приложений);

– *естественность описания* (использования естественного языка описания предметной области, легко выделяются объекты и связи между ними) [6].

Основные недостатки объектно-ориентированного подхода [6]:

– объектно-ориентированная декомпозиция заметно отличается от алгоритмической, что влечет за собой необходимость преодоления психологических трудностей и дополнительные расходы на обучение новой методике и языку программирования;

– основные недостатки объектно-ориентированного подхода лежат в области программирования: активное динамическое создание и удаление новых объектов (что широко используется в объектно-ориентированных языках);

– многочисленность методов и их излишние вызовы (для доступа к многим атрибутам объектов используются специальные методы).

Однако надо иметь в виду, что при использовании традиционных технологий некоторые результаты можно получить и при сравнительно небольших затратах, но на определенной стадии наступает насыщение, когда даже значительные дополнительные затраты не приводят к существенному повышению эффективности.

Контрольные вопросы

1. Дайте определение понятия «проектирование».
2. Дайте развернутую характеристику подходов к проектированию сложных технических систем.
3. Сформулируйте основные принципы проектирования.
4. Опишите укрупненную схему процесса проектирования сложных систем.

5. Назовите основные стадии и процедуры процесса проектирования.
6. Прокомментируйте содержание основных стадий процесса проектирования.
7. Прокомментируйте содержание основных процедур процесса проектирования.
8. Опишите назначение и порядок проведения предпроектной стадии проектирования.
9. Опишите блочно-иерархический подход к проектированию сложных систем (стили проектирования, аспекты описания).
10. Дайте характеристику структурному подходу к проектированию сложных систем (две части структурного подхода).
11. Укажите основные недостатки структурного подхода к проектированию.
12. Сформулируйте основные принципы необходимости объектно-ориентированного подхода к проектированию сложных систем.
13. Рассмотрите основные понятия объектно-ориентированного анализа (ООА).
14. Опишите фундаментальные понятия объектно-ориентированного подхода.
15. Сформулируйте основные принципы объектно-ориентированного подхода.
16. Укажите основные преимущества и недостатки объектно-ориентированного подхода.

Лекция 4

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ В МАШИНОСТРОЕНИИ

В лекции 1 (п. 1.1) было сказано, что в состав развитых машиностроительных систем автоматизированного проектирования (САПР) на этапах проектирования и подготовки производства входят в качестве составляющих системы CAD, CAE и CAM.

4.1. Общая характеристика систем автоматизированного проектирования

Рассмотрим основные функции автоматизированных систем CAD, CAE и CAM на этапах проектирования и подготовки производства.

CAD – автоматизированное проектирование – представляет собой компьютерную технологию для облегчения создания, изменения, анализа и оптимизации проектов. Наиболее важным компонентом CAD являются системы автоматизированной разработки рабочих чертежей и системы геометрического моделирования. Кроме того, CAD используется для оптимизации и анализа, имеет программы для анализа допусков, расчета массинерциальных свойств, моделирования методом конечных элементов и визуализации результатов анализа.

CAM – автоматизированное производство – это компьютерная технология для планирования, управления и контроля операций производства через прямой и косвенный интерфейс с производственными ресурсами предприятия. CAM используется для числового программирования управления станками с ЧПУ

и роботами, работающими на гибких автоматизированных участках. В задачах планирования производства САМ стыкуется с автоматизированной системой *MRP* и *ERP* через систему *PDM* (*MRP* – *material requirement planning* – планирование технических требований к материалам).

САЕ – автоматизированное конструирование – это компьютерная технология для анализа геометрии деталей, моделирования и изучения поведения изделий для усовершенствования и оптимизации их конструкций. Средства САЕ могут осуществлять множество различных вариантов анализа (кинематические расчеты, динамический анализ, расчет напряжений методом конечных элементов – *FEM* – *finite-element method*). Метод *FEM* рассчитывает теплообмен, распределение магнитных полей, потоков жидкости и другие задачи с непрерывными средами. Для применения метода *FEM* необходима аналитическая модель объекта подходящего уровня.

САЕ наиболее эффективно используется на стадии анализа и оптимизации конструкции.

Функции САД-систем

Функции САД-систем в машиностроении подразделяют на функции двумерного и трехмерного проектирования. К функциям *2D* относят черчение, оформление конструкторской документации; к функциям *3D* – получение трехмерных геометрических моделей, метрические расчеты, реалистичную визуализацию, взаимное преобразование *2D*- и *3D*-моделей. В ряде систем предусмотрено также выполнение процедур, называемых процедурами позиционирования. К ним относят компоновку и размещение оборудования, проведение соединительных трасс.

Среди САД-систем различают системы нижнего, среднего и верхнего уровней. Первые из них иногда называют «легкими» системами, они ориентированы преимущественно на *2D*-графику, сравнительно дешевы, основной аппаратной платформой для их использования являются персональные ЭВМ. Системы верхнего уровня, называемые также «тяжелыми», дороги,

более универсальны, ориентированы на геометрическое твердотельное и поверхностное 3D-моделирование, оформление чертежной документации в них обычно осуществляется с помощью предварительной разработки трехмерных геометрических моделей. Системы среднего уровня по своим возможностям занимают промежуточное положение между «легкими» и «тяжелыми».

К важным характеристикам САД-систем относятся параметризация и ассоциативность. Параметризация подразумевает использование геометрических моделей в параметрической форме, т.е. при представлении части или всех параметров объекта не константами, а переменными. Параметрическая модель, находящаяся в базе данных, легко адаптируется к разным конкретным реализациям и поэтому может использоваться во многих конкретных проектах. При этом появляется возможность включения параметрической модели детали в модель сборочного узла с автоматическим определением размеров детали, диктуемых пространственными ограничениями. Эти ограничения в виде математических зависимостей между частью параметров сборки отражают ассоциативность моделей.

Параметризация и ассоциативность играют важную роль при проектировании конструкций узлов и блоков, состоящих из большого числа деталей. Действительно, изменение размеров одних деталей оказывает влияние на размеры и расположение других. Благодаря параметризации и ассоциативности изменения, сделанные конструктором в одной части сборки, автоматически переносятся в другие части, вызывая изменения соответствующих геометрических параметров в этих частях.

САД-системы геометрического моделирования делятся на *каркасные, поверхностные, твердотельные и немногочисленные* [9].

Каркасное моделирование (wireframe modeling systems) представляют форму тела в виде набора характеризующих ее линий и конечных точек. Визуальная модель представляет собой каркасный чертеж формы, а соответствующее математическое описание представляет набор уравнений кривых, координат точек и сведений о связности кривых и точек (принадлежность точек

к конкретным кривым, а также пересечение кривых друг другом). Однако эта модель не содержит сведений о внутренних и внешних поверхностях объекта, что не позволяет рассчитать массу объекта.

В *системах поверхностного моделирования* (surface modeling systems) математическое описание визуальной модели включает в себя не только сведения о характеристических линиях и их конечных точках, как в каркасном моделировании, но и данные о поверхностях. При работе с отображаемой на экране моделью изменяются уравнения поверхностей, кривых и координат конечных точек. Существует три стандартных метода создания поверхностей в системах поверхностного моделирования:

- интерполяция входных точек;
- интерполяция криволинейных сеток;
- трансляция и вращение заданной кривой.

Системы твердотельного моделирования (solid modeling systems) предназначены для работы с объектами, состоящими из замкнутого объема или монолита (solid). Системы линий и поверхностей должны образовывать замкнутый объем, т.е. объем тела полностью заполнен материалом. Математическое описание объекта, созданного в системе твердотельного моделирования, содержит сведения, по которым система может определить, где находится какая-либо точка: внутри объекта, снаружи него или на его границе.

Немногообразные системы моделирования (nonmanifold modeling systems) позволяют использовать каркасные, поверхностные, твердотельные и сотовые (а также другие многообразные) модели одновременно в одной и той же системе моделирования, расширяя диапазон доступных моделей сверх возможностей любой из упомянутых систем. К сожалению, обычные системы геометрического моделирования (каркасные, поверхностные, твердотельные) не поддерживают моделирование многообразных моделей.

Более подробно системы геометрического моделирования описаны в [9].

Функции САМ-систем

Основные функции САМ-систем: разработка технологических процессов, синтез управляющих программ для технологического оборудования с ЧПУ, моделирование процессов обработки, в том числе построение траекторий относительного движения инструмента и заготовки в процессе обработки, генерация пост-процессоров для конкретных типов оборудования с ЧПУ, расчет норм времени обработки.

Другая важная функция САМ – программирование роботов, которые могут работать в гибких производственных системах и на автоматизированных участках. Еще одна важная функция САМ состоит в автоматизации *технологической подготовки производства* (Computer Aided Process Planning – CAPP). Автоматизация технологической подготовки производства связывает проектирование и производство и позволяет достигнуть полную интеграцию САД и САМ.

CAPP составляет основу САМ и выполняет синтез технологических процессов и программ для оборудования с ЧПУ, а также выбор оборудования, оснастки, инструмента, расчет норм времени и т. п.

Технология подготовки производства (CAPP) заключается в выборе технологических процессов и их параметров, а также оборудования для реализации этих процессов. Технологическую подготовку можно определить также, как составление подробных технологических инструкций для станка или сборщика агрегата, изделий.

Автоматизированный подход к подготовке производства реализуется в виде *модифицированного подхода* (variant approach) или *генеративного подхода* (generative approach).

Модифицированный подход называется так потому, что он является модификацией неавтоматизированного (традиционного) планирования производства, выполняемого вручную: технолог пользуется не только своей памятью, но и памятью компьютера. Другими словами, рабочий журнал технолога хранится в компьютерном файле.

Типичный технологический план производства подобной детали может автоматически извлекаться из такого файла после описания анализируемой детали в соответствии с определенной системой кодирования. Выбранный план производства может редактироваться в интерактивном режиме; в него вносятся поправки, соответствующие специфике конкретной детали. Таким образом, модифицированный подход требует наличия базы данных со стандартными планами производства для каждого семейства деталей. Такой план должен содержать все инструкции, которые будут входить в план производства любой детали из данного семейства. Детали классифицируются по семействам на основании концепции групповой технологии. Согласно этой концепции каждой детали присваивается код, зависящий от ее элементов, после чего детали группируются в семейства в соответствии с присвоенными кодами. О групповой технологии будет рассказано в другом разделе.

Итак, модифицированный подход к разработке плана производства выражается в следующем. Технологическая подготовка производства новой детали начинается с кодирования ее особенностей, что эквивалентно описанию детали на языке группой технологии. Затем деталь может быть отнесена к какому-либо семейству на основании ее кода, после чего из базы данных извлекается стандартный план производства для деталей этого семейства. В плане содержатся общие инструкции по производству любых деталей семейства, поэтому может потребоваться его редактирование для получения плана нужной детали.

Редактирование осуществляется средствами компьютерной системы. Часто изменения оказываются незначительными, потому что новый план представляет собой лишь небольшую модификацию стандартного. Благодаря этому на этапе подготовки плана экономится время, а готовые планы оказываются гораздо более последовательными, чем разрабатываемые вручную.

Если деталь нельзя отнести к одному из существующих семейств, технолог может разработать новый стандартный план производства в интерактивном режиме.

Генеративный подход (generative approach) состоит в том, что технологический план вырабатывается автоматически на основании технических требований к детали. В технические требования должны включаться подробные сведения о материале, особенностях обработки и предлагаемых методиках проверки, а также графическое изображение формы детали.

На первом этапе разработки плана производства новой детали при генеративном подходе технические требования вводятся в компьютерную систему. В идеале они должны считываться непосредственно из базы данных САПР. Для этого необходимо, чтобы автоматизированная система технологической подготовки могла распознавать элементы детали, требующие машинной обработки, такие как отверстия, пазы и выемки.

Реализация первого этапа значительно упрощается, если при моделировании детали используется объектно-ориентированный подход. Однако даже конструктивные элементы, используемые в системе объектно-ориентированного моделирования, могут потребовать преобразования к элементам, которые могут быть изготовлены машинной обработкой.

Некоторые конструктивные элементы однозначно сопоставляются с технологическими, тогда как преобразования других представляют собой не совсем обычную процедуру. Кроме того, информации об элементах, вообще говоря, недостаточно для технологической подготовки производства. Например, большинство моделей САД не содержат сведений о допусках и материалах, и их приходится вводить вручную. При этом схема кодирования должна определять все геометрические элементы и их параметры, в частности положение, размеры и допуски. Закодированные данные сопровождаются информацией в текстовом формате. Наконец, система должна иметь сведения о форме заготовки.

На втором этапе закодированные данные и текстовая информация преобразуются в подробный технологический план производства детали. На этом этапе определяется оптимальная последовательность операций и условия их выполнения. К условиям относятся используемые инструменты, крепления, измерительные приборы, зажимы, схемы подачи и скорости обработки. Для

построения столь подробного плана производства детали произвольной сложности требуется большая база данных и сложная логическая система.

Этап технологической подготовки производства в основном сводится к решению следующих задач [1]:

- разработка технологий изготовления изделий, инструментов, приспособлений и т. д. на основе их геометрических моделей, полученных на этапе проектирования (CAD);

- подготовка программ для технологического оборудования (станков, прессов, прокатных станов и др.) с ЧПУ по спроектированным технологиям.

Программные среды, с помощью которых решаются задачи этого этапа, можно объединить в две группы. К первой из них следует отнести программные комплексы, специально разработанные для выполнения всего цикла или определенных процедур технологической подготовки производства. Среди данной группы программного обеспечения можно выделить ADEM, ArtCAM, EdgeCAM и некоторые разработки российских фирм: КОМПАС, АВТОПРОЕКТ (Аскон) – проектирование технологических процессов механообработки, штамповки, сборки, термообработки; FLEX ТехноПро (Топ Системы) – проектирование технологии механообработки, сборки, сварки, пайки, нанесения покрытий, штамповки,ковки, термообработки; СИТЕП ЛШ – листовая штамповка; ТЕСНСАRD (Интермех) – комплексная система автоматизации технологической подготовки производства; ТехноПро (Вектор) – универсальная система автоматизации технологического проектирования; SprutCAM, СПРУТ-ТП (СПРУТ-Технологии) – система автоматизированного проектирования технологических процессов и др.

Другую группу программного обеспечения составляют программные системы сквозного проектирования и технологической подготовки производства: CATIA, EUCLID3, Unigraphics, Pro/Engineer, CADDS и др.

Контроль качества управляющих программ выполняют специальные программы, например такие, как NC Simul, NC Formater и др.

В производстве машиностроительных и части приборостроительных изделий используют технологии, в основе которых лежат различные физические процессы: механообработка, электроэрозийная обработка, литье металлов и пластмасс и др.

В автоматизированных системах сквозного проектирования и подготовки производства наиболее часто реализованы следующие виды механообработки: 2,5-, 3- и 5-координатное фрезерование, токарная обработка, сверление, нарезание резьбы и др.; имеется возможность моделировать движение инструмента и снятие материала во время черновой и чистовой обработки поверхности изделия. Например, в простейшем варианте 2- и 2,5-координатной обработки поверхностей: контурная обработка, фрезерование призм и тел вращения, выборка карманов с возможностью движения «в одну сторону», зигзаг, спираль, а также нарезание резьбы и снятие фасок. В модулях 3- и 5-координатного фрезерования программных систем сквозного проектирования и технологической подготовки производства реализованы практически все возможные способы обработки поверхностей изделий, например такие, как фрезерование поверхности с управлением угла наклона инструмента, шлифующее резание с возможностью обдувки и др.

При выполнении различных видов механообработки используется общая база данных для поддержки связи между геометрической моделью обрабатываемой детали и управляющей программой для станка с ЧПУ, где проходы инструмента создают по геометрии модели. Изменение геометрии отражается в управляющей программе. Траектория движения инструмента создается интерактивно по поверхности модели изделия, благодаря чему технологи получают возможность визуально наблюдать на экране монитора имитацию процесса удаления стружки, контролировать зарезы и быстро вносить изменения в циклы обработки.

С помощью специальных функций автоматически вычисляется объем, который необходимо удалить из заготовки при обработке изделия.

В системах сквозного проектирования и технологической подготовки производства различных фирм-разработчиков реализо-

ваны свои подходы к организации программ, свои алгоритмы и методы. Более подробно эти подходы реализованы в системах верхнего уровня САПР, например в системе EUCLID3 фирмы EADS MATRA DATA VISION (Франция), которая воплотила многолетний опыт специалистов различных отраслей и считается одной из лучших в мире. В [1] описаны основные положения проектирования технологий фрезерной и токарной обработки.

Практика показывает, что предприятия по-разному подходят к созданию своих интегрированных информационных систем, предназначенных для комплексного решения задач автоматизации конструирования, инженерного анализа и технологической подготовки производства. Во многих случаях на рабочих местах конструкторов и технологов устанавливаются программные среды различных фирм-разработчиков. В этих условиях вопросы организации обмена информацией становятся актуальными. Известно, что обмен без потерь информации достигается при наличии единой базы данных для различных подсистем. Этим выгодно отличаются комплексные системы сквозного проектирования и подготовки производства верхнего уровня. В тех случаях, когда на рабочих местах устанавливаются программные среды различных фирм, организация обмена информацией ложится на самих пользователей. Поэтому важно, чтобы для таких программ были разработаны соответствующие интерфейсы с необходимой полнотой реализации форматов (CALS-технологии).

При подготовке производства также широко используется *групповая технология проектирования и подготовки производства*.

Групповая технология – это группировка задач, в основном подобных друг другу, позволяющая найти общее их решение, сэкономив тем самым время и усилия.

Это всеобъемлющее определение можно конкретизировать, определив область применения. Суть групповой технологии состоит в создании базы данных подобных деталей, проектов и технологий и использовании этой базы для внедрения общей процедуры проектирования и производства таких деталей. Детали объединяются в семейства по конструктивному подобию

(например, по схожести форм) и по технологическому подобию (например, по необходимым операциям, таким как фрезерование или сверление).

Групповая технология широко используется для упрощения продвижения продуктов на производстве. Выделение деталей с общими технологическими параметрами позволяет разработать эффективные планы производства, выделяя для каждого семейства одну ячейку плана. Таким образом, упрощаются технологические маршруты, сокращаются временные затраты на передачу материалов между станками и длительность производственных циклов.

Более того, поскольку подобные детали производятся на одних и тех же станках, часто сокращаются и временные затраты на настройку станков. Может использоваться специальное технологическое оснащение. В предыдущем разделе отмечалось, что коды групповой технологии используются для выбора существующих планов в автоматизированной технологической подготовке производства.

Инженерам-технологам не приходится разрабатывать планы с нуля для каждой новой детали – вместо этого они могут обращаться к планам производства аналогичных деталей и изменять их в соответствии с техническими требованиями к новым деталям.

Концепция групповой технологии дает преимущества и на этапе проектирования. С ее помощью часто удается устранить избыточное разнообразие деталей, предоставив конструкторам возможность осуществлять поиск по семействам деталей. Часто конструкторы просто не знают о наличии аналогичных проектов среди текущих разработок. Обычно это бывает связано с тем, что система нумерации деталей не дает им достаточной информации. В таких случаях возникает тенденция к дублированию деталей с незначительными изменениями, не имеющими отношения к назначению деталей. Избыток деталей приводит к быстрому увеличению количества бумаг, а также к расходам заготовок.

В процессе разработки технических изделий широкое применение находят их физические прототипы [9].

Быстрое прототипирование и изготовление (БПИ) – это способ физического моделирования для получения физического прототипа конструкции, например детали (физический объект), – широко используется на стадии анализа проекта (детали) по окончании концептуального проектирования.

В своей основе процессы быстрого прототипирования и изготовления (БПИ) (*rapid prototyping and manufacturing*) состоят из трех шагов: формирование поперечных сечений изготавливаемого объекта из полимерных материалов (или за счет спекания твердых частиц и порошка лучом лазера); послойное наложение этих сечений и комбинация слоев. Поэтому для создания физического объекта требуются только данные о его поперечных сечениях. Процессы БПИ являются безынструментальными – позволяют создавать физический объект без использования инструментов (по сравнению, например, со станками с ЧПУ).

Данные о поперечных сечениях изделия можно получить, используя объемные 3D-чертежи деталей, полученных методами геометрического (например, твердотельного) моделирования.

Методы БПИ [9, с. 379–412]

1. Стереолитография.
2. Отверждение на твердом основании.
3. Избирательное лазерное спекание.
4. Трехмерная печать.
5. Ламинирование.
6. Моделирование методом наплавления.

Кратко рассмотрим все перечисленные методы.

Стереолитография. Метод основан на избирательном отверждении поверхностного слоя фотополимера и построении трехмерных объектов из последовательно наложенных слоев.

Процесс изготовления детали протекает и следующим образом (рис. 4.1):

1. Фоточувствительный полимер, затвердевающий на свету, поддерживается в жидком состоянии.
2. На толщину одного слоя ниже поверхности жидкого полимера помещается платформа, способная двигаться в вертикальном направлении.

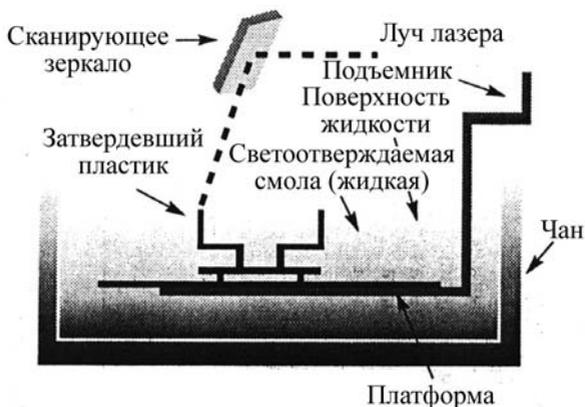


Рис. 4.1. Процесс стереолитографии

3. Ультрафиолетовый лазер сканирует слой полимера над платформой, отверждая полимер по форме соответствующего поперечного сечения. Обратите внимание, что этот процесс начинается с нижнего поперечного сечения детали.

4. Платформа опускается в ванну с полимером на толщину одного слоя, давая полимеру растечься по поверхности детали для начала нового слоя.

5. Шаги 3 и 4 повторяются до тех пор, пока не будет наращен верхний слой детали.

6. Для полного затвердевания детали выполняется окончательное отверждение. Этот шаг необходим, поскольку в каждом слое могут еще оставаться жидкие участки. Так как лазерный луч имеет конечные размеры, сканирование каждого слоя аналогично закрасиванию некоторой фигуры тонкой цветной ручкой.

Габаритные размеры ванн $500 \times 500 \times 500$ мм. Толщина слоев составляет $50 \dots 150$ мкм. Точность моделей $0,1$ мм.

Отверждение на твердом основании (*solid ground curing – SGC*). В этом случае каждый слой отверждается путем экспонирования ультрафиолетовой лампой, а не сканирования лазерным лучом. Таким образом, все точки слоя затвердевают одновременно и окончательного отверждения не требуется. Типичный

процесс отверждения на твердом основании имеет место в системе *Solider* от *Cubatal Israel*, работа которой происходит следующим образом:

1. По данным геометрической модели детали и желаемой толщине слоя рассчитывается поперечное сечение каждого слоя.

2. Для каждого слоя изготавливается оптическая маска по форме соответствующего поперечного сечения.

3. После выравнивания (рис. 4.2, *a*) платформа покрывается тонким слоем жидкого фотополимера (рис. 4.2, *б*).

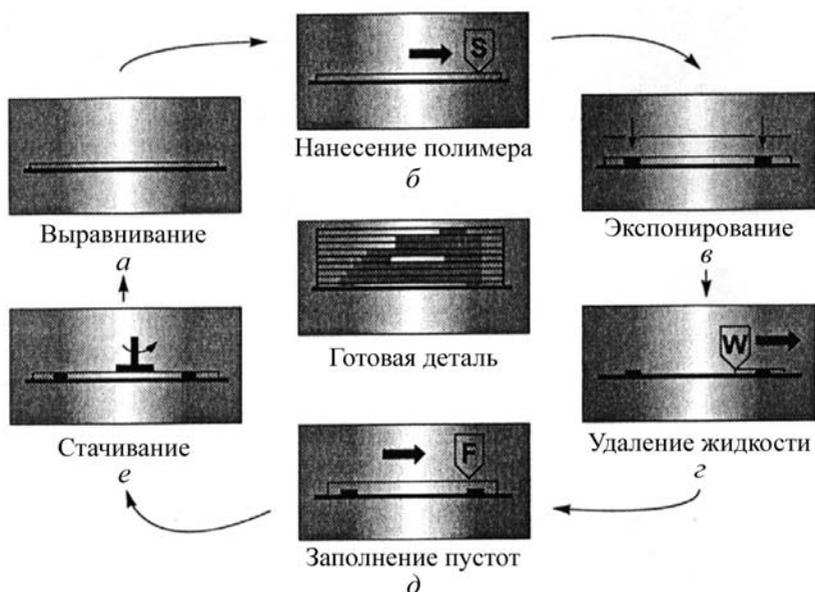


Рис. 4.2. Система Solider

4. Над поверхностью жидкой пластмассы помещается маска, соответствующая текущему слою, и пластмасса экспонируется светом мощной ультрафиолетовой лампы (рис. 4.2, *в*). Обратите внимание, что процесс начинается с маски, соответствующей нижнему слою.

5. Оставшаяся жидкость удаляется с изделия аэродинамическим *wiper* [‘waipɐ] – очистителем (рис. 4.2, *г*).

6. Изделие покрывается слоем жидкого воска, который заполняет пустоты (рис. 4.2, д). Затем к воску прикладывается холодная пластина, и он затвердевает.

7. Слой стачивается до желаемой толщины с помощью шлифовального диска (рис. 4.2, е).

8. Готовая часть изделия покрывается тонким слоем жидкого полимера, и шаги 4–7 повторяются для каждого последующего слоя, пока не будет обработан самый верхний слой.

9. Воск расплавляется и удаляется из готовой детали.

Главным преимуществом метода отверждения на твердом основании по сравнению с методом стереолитографии является отсутствие необходимости в поддерживающих структурах. Это обусловлено тем, что пустоты заполняются воском. Кроме того, благодаря использованию света лампы вместо лазерного луча, исключается операция окончательного отверждения. Хотя отверждение на твердом основании позволяет изготавливать детали с большей точностью, чем стереолитография, процесс этот весьма сложен.

Избирательное лазерное спекание. Процесс изготовления детали путем избирательного лазерного спекания, разработанный фирмой DTM (США), протекает следующим образом (рис. 4.3):

1. Цилиндрическая заготовка помещается на высоте, необходимой для того, чтобы на нее можно было осадить слой порошкового материала желаемой толщины. Порошковый материал, используемый для изготовления прототипа, поступает из подающего цилиндра и наносится выравнивающим валиком.

2. Слой порошка избирательно сканируется и нагревается лучом лазера, вследствие чего частицы слипаются между собой. Просканированные частицы порошка образуют требуемое поперечное сечение. Обратите внимание, что этот процесс начинается с нижнего поперечного сечения детали.

3. Цилиндрическая заготовка опускается на толщину одного слоя для нанесения нового слоя порошка.

4. Луч лазера сканирует новый слой порошка, склеивая его с предыдущим и формируя следующее поперечное сечение.

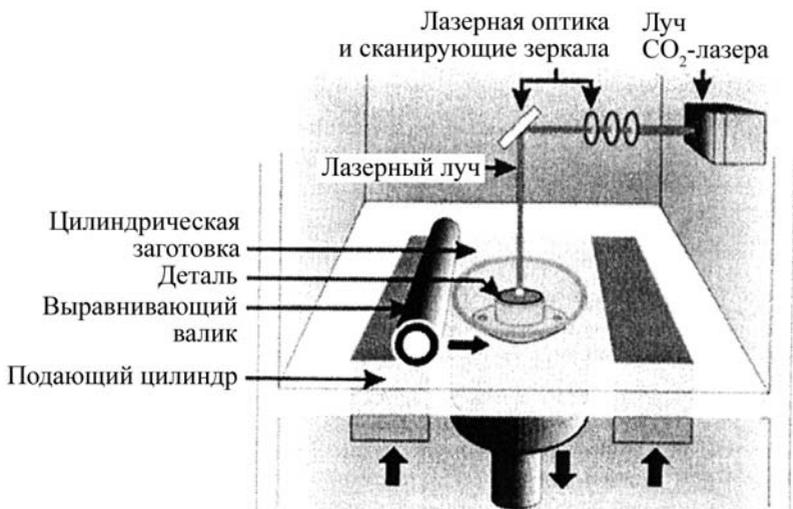


Рис. 4.3. Избирательное лазерное спекание

5. Шаги 3 и 4 повторяются до тех пор, пока не будет создан самый верхний слой детали.

6. Для некоторых материалов может понадобиться окончательное отверждение.

Поддерживающая структура не требуется, потому что пустоты каждого слоя заполняются необработанным порошком. Более того, в качестве материала для процесса избирательного спекания потенциально может использоваться любой плавкий порошок, даже металлический, если лазер обладает достаточной мощностью.

На практике для металлических порошков, частицы которых покрыты термопластическим связующим материалом, используется косвенное спекание. Под лучом лазера связующий материал расплавляется и свободно связывает частицы металлического порошка, образуя желаемую форму, которая называется *зеленой деталью* (*green part*). В этом случае достаточно, чтобы мощности лазера хватало для расплавления связующего материала.

Затем зеленая часть подвергается обработке в печи, в ходе которой связующий материал выжигается, а частицы металлического

порошка связываются за счет обычных механизмов спекания. Получившаяся деталь носит название *коричневой детали (brown part)*. Без дальнейшей обработки деталь будет довольно пористой из-за наличия пустот, которые ранее занимали частицы связующего материала. Чтобы снизить пористость, в печь помещается еще один материал – инфильтрант. Этот металл расплавляется при рабочей температуре печи и проникает в поры детали за счет капиллярного эффекта. Данный метод используется для изготовления форм для литья непосредственно по их геометрическим моделям. Ресурса таких форм достаточно для изготовления от 2 500 до 10 000 деталей-прототипов.

Трехмерная печать. Разработанный в Массачусетском технологическом институте процесс трехмерной печати был назван так из-за своей схожести с печатью на струйном принтере. В трехмерной печати вместо чернил используется жидкое связующее вещество. Процесс трехмерной печати происходит следующим образом (рис. 4.4):

1. Платформа располагается на высоте, необходимой для того, чтобы можно было нанести на нее слой керамического порошка надлежащей толщины.

2. Нанесенный слой керамического порошка избирательно сканируется печатающей головкой, из которой поступает жидкое связующее вещество, вызывающее прилипание частиц друг к другу. Отсканированные печатающей головкой частицы образуют требуемую форму поперечного сечения. Обратите внимание, что этот шаг начинается с нижнего поперечного сечения.

3. Платформа опускается на одну толщину слоя, позволяя нанести следующий слой порошка.

4. Новый слой сканируется, образуя следующее поперечное сечение и склеиваясь с предыдущим слоем.

5. Шаги 3 и 4 повторяются, пока не будет создан верхний слой детали.

6. Для отверждения детали проводится последующая тепловая обработка.

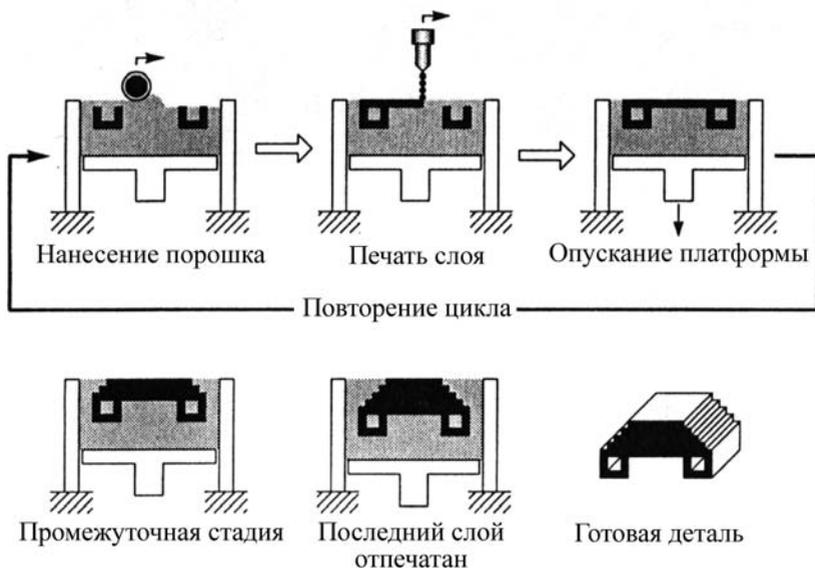


Рис. 4.4. Трехмерная печать

С помощью трехмерной печати было бы удобно изготавливать формы для литья, поскольку форма изготавливается как цельная деталь, состоящая из оболочки и полостей, и положение полостей относительно оболочки можно было бы задавать точно. Однако формы для литья, изготовленные путем современной трехмерной печати, имеют невысокое качество поверхности.

Ламинирование (*laminated object manufacturing – LOM*). Коммерциализированной фирмой *Helisys* деталь изготавливается путем ламинирования и лазерной резки материалов, поступающих в листовом виде. Слипание листов происходит за счет наличия термоадгезивного покрытия. Процесс протекает следующим образом (рис. 4.5):

1. Каждый лист приклеивается к заготовке с помощью нагрева и давления, образуя очередной слой. Листовой материал подается в виде непрерывного рулона с одной стороны машины и принимается с противоположной стороны. Температуру и давление,

необходимые для ламинирования, обеспечивает нагретый валик. Обратите внимание на то, что, когда к стопке приклеивается следующий лист, платформа опускается на толщину одного листа.

2. После того как слой (лист) приклеен, он сканируется лазером вдоль контура текущего поперечного сечения. Обычно для этой цели используется лазер на углекислом газе мощностью 25 или 50 Вт. Как и в других процессах, этот шаг начинается с нижнего поперечного сечения. Обратите внимание, что здесь сканирование производится только по контурам. Это делает данный процесс более эффективным, чем процессы, требующие растрового сканирования.

3. Области слоя, выходящие за пределы контуров, штрихуются лазером, т. е. рассекаются на маленькие кусочки, называемые *черепичками (tiles)*, для последующего удаления, когда деталь будет закончена.

4. Шаги 1–3 повторяются до тех пор, пока не будет наклеен и вырезан верхний слой детали.

5. После того как все слои будут готовы, внутри блока поддерживающего материала окажется готовая деталь. Этот материал затем разматывается на кусочки вдоль линий лазерной штриховки.

6. Готовую деталь можно покрыть герметиком, чтобы предохранить ее от влажности.

Наличие поддерживающего материала вокруг детали имеет свои преимущества и недостатки. Прежде всего, оно исключает необходимость во внешних поддерживающих структурах. При изготовлении детали внутри блока поддерживающего материала, имеющего определенную форму, геометрия всей структуры стабилизирована и соответственно ей не грозит перекося под собственным весом. Более того, нам не приходится беспокоиться об изолированных «островках», которые часто образуются, когда твердое тело, спроектированное в CAD-системе, рассекается на слои.

Иными словами, ламинирование позволяет избежать создания специальных подпорок, которые точно фиксировали бы эти «островки» в пространстве, пока в процессе изготовления не будут созданы «мосты» к оставшимся частям детали. Однако удаление лишнего материала по окончании изготовления детали является непростой задачей.

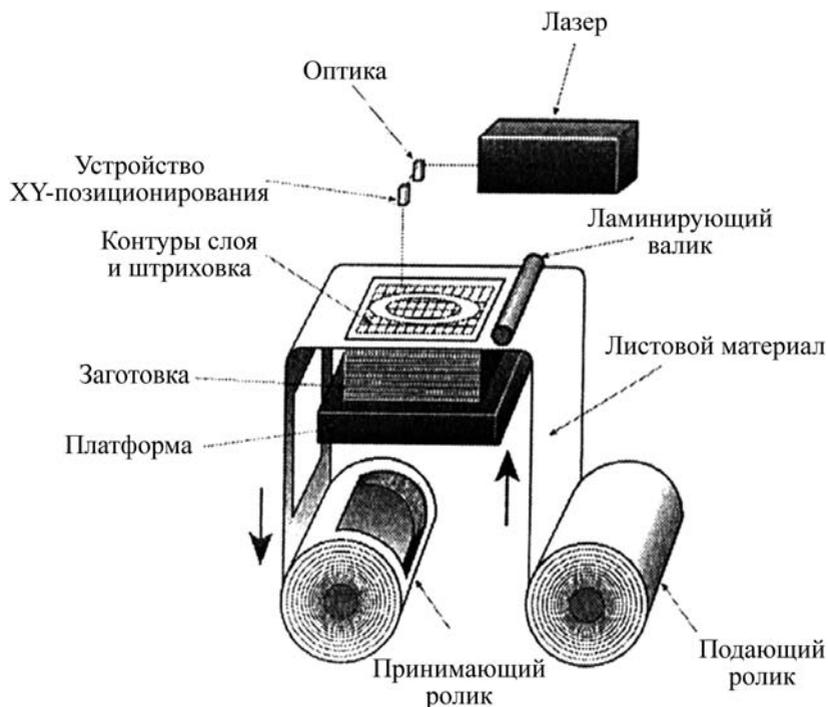


Рис. 4.5. Процесс ламинирования

Чтобы удалить только излишки, а хрупкие части детали при этом не сломать, необходимо бережно очистить ее вручную. Кроме того, полую структуру с замкнутыми поверхностями невозможно изготовить в виде единой части, поскольку в этом случае излишки материала невозможно будет извлечь изнутри. Сложность удаления ненужного материала характерна для любой части с узкими перемычками, внутренними полостями с ограниченным доступом, слепыми отверстиями и т.п. Далее большая часть материала, расходуемая при ламинировании, идет не на саму деталь, а остается неиспользованной в рулоне или образует поддерживающие структуры, которые будут удалены после изготовления. Это может быть весьма расточительно, если применяются более дорогостоящие материалы, чем бумага.

Помимо преимуществ и недостатков процесс ламинирования имеет следующие характеристики:

Это субтрактивный, а не аддитивный процесс, т. е. для создания слоя с требуемым поперечным сечением материал удаляется, а не добавляется. Во всех прочих процессах БПИ слои образуются путем добавления материала. Таким образом, потенциально ламинирование является самой быстрой технологией изготовления деталей с большим отношением объема к площади поверхности.

Детали образуются перемежающимися слоями материала и клеящего вещества. Соответственно, многие из их физических свойств являются неоднородными и анизотропными.

Потенциальная точность процесса изготовления ламинированных объектов высока. В нем может использоваться сколь угодно тонкий листовый материал, что позволяет достичь хорошей разрешающей способности в направлении наращивания деталей. В действительности изготовить тонкий однородный листовый материал несложно, и усадка при ламинировании не представляет проблемы, поскольку контуры вырезаются после того, как происходит усадка.

Хотя процесс потенциально применим ко многим материалам, включая пластики, композиты и металлы, наиболее популярным на сегодняшний день является бумажное ламинирование.

Габариты изделия достигают размера $800 \times 560 \times 500$ мм. Масса до 20 кг. Температура горячего валика – $70 \dots 400$ °С. Сила прижатия валика до 180 кН. Толщина материала $0,076 \dots 0,2$ мм.

Моделирование методом наплавления (*fused-deposition modeling* – FDM). В процессе наплавления коммерциализированной фирмой *Stratasys* каждый слой формируется путем выдавливания термопластичного материала, находящегося в жидком состоянии (рис. 4.6). Температура выдавливаемого материала незначительно превышает его температуру затвердевания (аналогичная ситуация при создании надписей на торте шоколадным кремом). Деталь изготавливается путем последовательного наплавления слоев. Этот процесс относительно прост, но его применение ограничено термопластичными материалами.

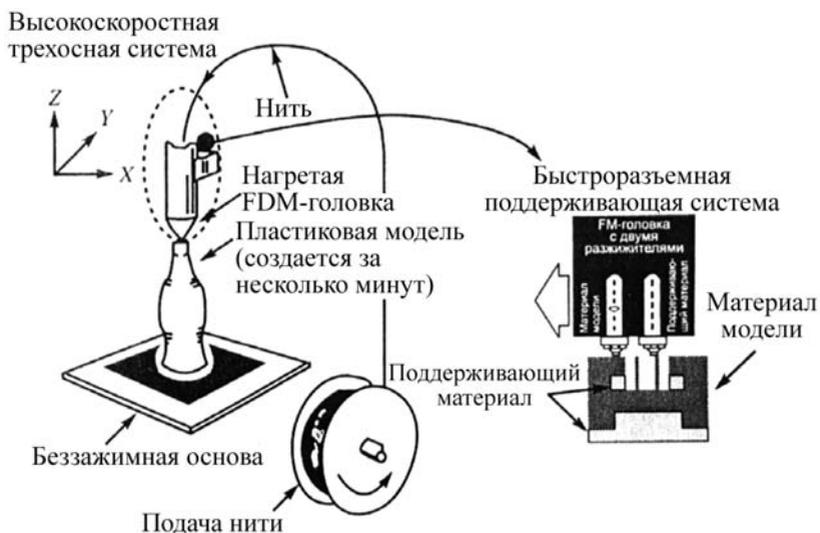


Рис. 4.6. Моделирование методом наплавления (с разрешения Stratasys, Inc.)

Функции САЕ-систем

САЕ-системы наиболее эффективно используются на стадии анализа и оптимизации конструкций и технологических процессов.

Функции САЕ-систем довольно разнообразны, так как связаны с проектными процедурами анализа, моделирования, оптимизации проектных решений. В состав машиностроительных САЕ-систем прежде всего включают программы для выполнения следующих процедур:

- моделирование полей физических величин, в том числе анализ прочности, который чаще всего выполняется в соответствии с МКЭ;
- расчет состояний моделируемых объектов и переходных процессов в них средствами макроуровня;
- имитационное моделирование сложных производственных систем на основе моделей массового обслуживания и сетей Петри.

Основными частями программ анализа с помощью МКЭ являются библиотеки конечных элементов, препроцессор, решатель и постпроцессор.

Библиотеки конечных элементов содержат их модели – матрицы жесткости. Очевидно, что модели конечных элементов будут различными для разных задач (анализ упругих или пластических деформаций, моделирование полей температур, электрических потенциалов и т. п.), разных форм конечных элементов (например, в двумерном случае – треугольные или четырехугольные элементы), разных наборов координатных функций.

Исходные данные для препроцессора – геометрическая модель объекта, чаще всего получаемая из подсистемы конструирования. Основная функция препроцессора – представление исследуемой среды (детали) в сеточном виде, т. е. в виде множества конечных элементов.

Решатель – программа, которая ассемблирует (собирает) модели отдельных конечных элементов в общую систему алгебраических уравнений и решает эту систему одним из методов разреженных матриц.

Постпроцессор служит для визуализации результатов решений в удобной для пользователя форме. В машиностроительных САПР это графическая форма. Пользователь может видеть исходную (до нагружения) и деформированную формы детали, в которых палитра цветов или интенсивность свечения характеризуют значения фазовой переменной.

САЕ-системы позволили создать новые методы визуализации результатов компьютерного моделирования отдельных прототипов и целых производств [9].

Дело в том, что прогресс автоматизированной разработки принес новую парадигму в проектирование и анализ. Прежде всего, системы геометрического моделирования настолько продвинулись вперед за последнее десятилетие, что современные САД-системы способны обрабатывать модели деталей и агрегатов самой сложной геометрии и конструкции. Агрегат можно отображать, оценивать и модифицировать как единое целое, а его движение имитировать так же, как это делается с физическим прототипом.

Еще одно достижение – это анализ по методу конечных элементов. Он мог бы стать средством виртуальной оценки надежности и технических характеристик продукта, если бы его вычислительную эффективность можно было улучшить до такой степени, чтобы выводить результаты в реальном времени. С его помощью можно было бы предсказывать механические свойства и характеристики (напряжения, прогиб, вибрация, температура, давление в сложных деталях) так, как если бы они измерялись посредством различных экспериментов.

Более того, производители САD-систем в настоящий момент пытаются объединить геометрическое моделирование с методом конечных элементов. Такая интеграция обеспечила бы непрерывное течение цикла проектирования и анализа. Метод конечных элементов использовался бы изначально в процессе проектирования для принятия конструкторских решений, что позволило бы сэкономить время и затраты, связанные с перепроектированием.

Все эти тенденции привели к возникновению новой инженерной дисциплины – *виртуальная инженерия (virtual engineering)*.

Виртуальная инженерия – это имитационный метод, помогающий инженерам в принятии решений и управлении. Виртуальная среда представляет собой вычислительную структуру, позволяющую точно имитировать геометрические и физические свойства реальных систем. Виртуальная инженерия включает в себя имитацию различных видов инженерной деятельности, таких, как машинная обработка, сборка, управление производственными линиями, осмотр и оценка, а также процесс проектирования.

Таким образом, виртуальная инженерия может охватывать весь цикл разработки и производства продукта. После того как смоделирована деталь, имитируется ее машинная обработка и сборка. Затем, также с помощью имитации, собранный прототип тестируется, и в его конструкцию вносятся необходимые изменения. Когда прототип одобрен, имитируется производственная система и ее функционирование. Прогнозируется также себестоимость и график поставок. В результате этих имитаций

получается оптимизированный конечный прототип и производственные процедуры, на основе которых затем реализуется физическая система.

Виртуальная инженерия дает совершенно новый подход к инженерным задачам. Использование имитации устранил необходимость в дорогостоящих физических прототипах и физических экспериментах. Время разработки коренным образом сократится, появится возможность проверить большее количество альтернативных вариантов конструкции, повысится качество конечного продукта. Виртуальная инженерия обеспечит также превосходный интерфейс для клиента, позволяя ему заранее увидеть трехмерную модель продукта и запросить конструктивные изменения. Можно будет построить прототип продукта, который недоступен, слишком опасен или слишком дорог для того, чтобы создавать его в реальности. Такая возможность будет неоценима в автомобильной и авиационной промышленности, где физические макеты стоят дорого, время разработки велико, продукты крайне сложны и требуется глубокая обратная связь от клиентов.

Основными компонентами виртуальной инженерии являются: *виртуальное проектирование, цифровая имитация, виртуальное прототипирование и виртуальный завод*. Виртуальное проектирование выполняется с помощью устройств виртуальной реальности. Цифровая имитация позволяет проверять и оценивать работу продукта без использования физических прототипов. В процессе виртуального прототипирования строится компьютерный прототип, имеющий ту же геометрию и физическое поведение, что и реальный продукт. Виртуальный завод – это имитация заводской производственной линии. Ниже следует описание каждой из этих составляющих.

Виртуальное проектирование выполняется в виртуальной среде с использованием технологий виртуальной реальности. Виртуальное проектирование сосредоточивается на альтернативном пользовательском интерфейсе для процесса проектирования. Используя технологии виртуальной реальности, конструкторы могут погрузиться в виртуальную среду, создавать компоненты, модифицировать их, управлять различными

устройствами и взаимодействовать с виртуальными объектами в процессе конструкторской деятельности. Конструкторы могут видеть стереоскопическое изображение виртуальных объектов и слышать пространственный реалистичный звук. Эти изображения и звук возникают, когда рука конструктора движет виртуальной рукой и пальцем. Прикосновение к виртуальному объекту ощущается конструктором в виде обратной связи. Тем самым замысел конструктора эффективно воплощается в проекте и проверяется функциональное поведение конструкции.

Основная цель виртуального проектирования – позволить конструктору действовать интуитивным и естественным образом. В системах геометрического моделирования, даже при том что современные CAD-системы предоставляют изощренные средства моделирования, взаимодействие конструктора с моделью неограничено. Возможности обзора ограничиваются изображением, спроектированным на монитор, а возможности ввода информации от конструктора – точечными манипуляциями с мышью.

Таким образом, в рамках CAD-технологии сегодняшнего дня конструктор является «одноглазым и однопалым». Более естественное взаимодействие, обеспечиваемое технологиями виртуальной реальности, дало бы конструктору большую свободу и позволило бы ему повысить креативность при создании модели. Вторая цель виртуального проектирования – на ранних стадиях проектирования учесть точку зрения потенциального пользователя продукции.

В процессе проектирования могут быть в полной мере оценены такие качества, как доступность и управляемость. Третья цель – учесть при проектировании опыт экспертов в сборке или манипулировании деталями. Этот опыт сложен и трудно формализуем, но система виртуального проектирования может пролить свет на положение пользователя, его взаимодействие с объектами и последовательность операций сборки.

Виртуальное проектирование потребует совершенно иного подхода к моделированию трехмерной геометрии. Например, меню и кнопки можно заменить технологией распознавания речи

или жестов. Если конструктор захочет изменить размеры модели, то, вместо того чтобы указать на нее, конструктор сможет взять ее в руки и растянуть. В связи с процессом виртуального проектирования непременно возникнут новые методы проектирования и схемы моделирования.

Цифровая имитация используется для проверки процессов, так как машинные операции необходимо тщательно проверить, прежде чем начинать реальную работу. Если в управляющем коде имеется ошибка, это может привести к серьезной поломке станка. Используя цифровую имитацию, пользователь перед началом работы имеет возможность проверить траекторию перемещения инструмента станка с ЧПУ, щупа координатно-измерительной машины или руки робота. Например, процессом машинной обработки можно управлять графически, как если бы это делалось на реальном станке: виртуальный станок с ЧПУ, читая стандартные инструкции, будет выполнять всю обработку в реальном времени, включая перемещение инструментов, принадлежностей, укладчиков и деталей. Непрерывный контроль за снятием материала позволяет пользователю выявлять ситуации, приводящие к вибрации и поломке инструмента, появлению выбоин и зарубок. С помощью имитации пользователь может также спрогнозировать столкновения между инструментом и приспособлением или деталью.

Визуализация также помогает инженерам лучше понять систему. Она позволяет легко уяснить идею конструкции и заранее проверить ее эксплуатационные качества. В настоящее время для этой цели используется главным образом кинематическая имитация твердых тел. Имитация моделей более высокого уровня – жидкостей, человеческих существ и сложных сред – требует моделирования физических эффектов, включая эффекты динамики, вибрации, акустики и деформации. Однако сложные имитации с использованием виртуальных прототипов могут осуществить проверку рабочих характеристик системы быстрее и с меньшими затратами.

Виртуальным прототипированием (*virtual prototyping*) называют построение прототипа агрегата из геометрических моделей его отдельных частей. Виртуальный прототип называют

иногда *цифровым макетом (digital mockup)* или *предварительной цифровой сборкой (digital preassembly)*. Системы цифрового прототипирования позволяют визуализировать процесс сборки и проверять осуществимость предлагаемых агрегатов в рамках имеющихся производственных ограничений. Путем сборки виртуального прототипа можно выявить конструктивные просчеты и внести изменения в проект, чтобы реальную сборку можно было выполнить с первой попытки.

Основная функция виртуального прототипирования заключается в том, чтобы установить, осуществима ли операция сборки. Система проверяет сопряжение деталей в контексте сборочных ограничений и требований к допускам. Функции обнаружения столкновений указывают на мешающие друг другу детали. Проверяется также последовательность сборки и траектории движения деталей. Кроме того, можно определить оптимальные траектории сборки. Передовые системы позволяют также проводить структурный и функциональный анализ виртуального прототипа, используя встроенное аналитическое программное обеспечение. Часто выполняется кинематическая и динамическая имитация прототипа.

Инженеры смогут основывать конструкторские решения на виртуальных прототипах. Оптимизация конструкции будет достигаться путем последовательного уточнения виртуального прототипа. С увеличением степени детальности виртуального прототипирования станет возможной более точная структурная и функциональная имитация. Идеальной ситуацией станет тогда, когда виртуальный прототип будет иметь всю совокупность характеристик физического прототипа. Таким образом, со временем виртуальное прототипирование устранил необходимость в дорогих и трудоемких физических прототипах.

Виртуальный завод (virtual factory) – это смоделированная на компьютере полная производственная система. Виртуальный завод имитирует конструкции производственных участков, производственные процессы и складские системы. Кроме того, для него можно программировать автоматизированное заводское оборудование – роботы, конвейеры и станки.

Система моделирует и имитирует такие компоненты, как конвейеры, накопители, доки, производственные участки и процессы. Кроме того, система моделирует процедуры – маршруты, последовательности и слияния. После этого модель производства анализируется в терминах затрат на рабочую силу, инвентаризации, эксплуатационных расходов, затрат на обработку и длительностей цикла.

Такие возможности позволяют использовать виртуальный завод для планирования производства, включая оценку проектов производственных систем и сравнение альтернативных способов производства. Когда эта технология достигнет зрелости, с помощью виртуального завода можно будет симитировать всю цепочку поставок, что позволит оценивать и оптимизировать весь процесс управления ресурсами и производством.

Итак, методы виртуальной инженерии предлагают совершенно новый подход к процессу проектирования, который позволяет оценить возможность производства различных вариантов конструкции (включая оценку качества сборки или эксплуатационную характеристику продукта) и оптимизировать производственный процесс (методом цифровой имитации), легко настроить продукт под требования заказчика, эффективно накопить обширную базу знаний, обеспечить основу для коллективной разработки проектов.

Рассмотрим аппаратуру, которая используется виртуальной инженерией. Виртуальная инженерия – чисто программная технология, и поэтому сама по себе не требует какого-либо специального оборудования. Однако для взаимодействия с пользователем необходимо оборудование виртуальной реальности. Это оборудование включает в себя устройства как ввода, так и вывода. Устройства вывода дают пользователю ощущения от виртуальной среды. Поскольку самый эффективный способ сенсорного восприятия – это зрение, главными компонентами систем виртуальной реальности являются устройства отображения. Эти устройства должны обеспечивать пользователю стереоскопический обзор.

Доступное в настоящий момент оборудование включает в себя головные дисплеи, биноккулярные всенаправленные мониторы,

дисплеи пространственного погружения и специальные очки. Звук и осязание обогащают ощущения от виртуальной реальности, когда они используются в совокупности со зрительной системой. Типичным примером звуковой аппаратуры могут служить наушники с пространственно расширенной звуковой системой. Аппаратура осязания – это приборы с силовой обратной связью. Популярными устройствами ввода являются системы распознавания речи, следящие системы и информационные перчатки.

Головные дисплеи. Головной дисплей – это устройство отображения с полным погружением. Шлем полностью закрывает глаза и позволяет смотреть только прямо перед собой. Небольшой экран, смонтированный перед глазом пользователя, дает стереоскопическое изображение. Прибор имеет встроенную следящую систему, благодаря которой изображение на дисплее меняется при изменении положения и ориентации головы пользователя. В настоящее время доступно более 40 моделей головных дисплеев. Среди них можно упомянуть FOHMD от CAE-Electronics и Looking Glass от Polhemus Lab.

Бинокулярные всенаправленные мониторы. Бинокулярный всенаправленный монитор – это механическая версия головного дисплея. Он состоит из дисплейной коробки, которая сбалансирована по весу на многозвенной стреле. Пользователь смотрит на дисплей, а движение его головы отслеживается через систему механических звеньев. Основным преимуществом бинокулярных всенаправленных мониторов перед головными дисплеями является быстрое и точное слежение. В продаже имеются бинокулярные всенаправленные мониторы BOOM-2C от Fake Space Labs (<http://www.fakespace.com>) и Cyberface3 от LEEP.

Дисплеи пространственного погружения. В дисплеях пространственного погружения используется панорамный видеоэкран, окружающий пользователя, так что пользователь чувствует себя погруженным в виртуальную среду. Дисплеи пространственного погружения обеспечивают большое поле зрения и свободу передвижения в виртуальной среде. В университете штата Иллинойс (Чикаго) разработана четырехстенная система пространственного погружения CAVE. Приобрести систему

CAVE можно через фирму *Pyramid System*. Существуют большие куполообразные дисплеи пространственного погружения, такие как *Visionarium Graphics*, имеющий 8 м в диаметре. Фирма *Spitz Electrihorizen* построила дисплей пространственного погружения диаметром 8,5 м.

Шторные очки. Шторные очки – это недорогое устройство отображения. Пользователь надевает устройство, напоминающее очки, которое попеременно закрывает обзор то одному, то другому глазу. Монитор или другое устройство отображения, синхронизированное с очками, имеет в два раза большую частоту обновления и попеременно показывает картинку для левого и для правого глаза. В результате на экране монитора пользователь видит стереоскопическое изображение. Данное устройство может использоваться вместе с выпускаемыми в настоящий момент дисплеями, поэтому оно экономически выгодно. Однако оно не обеспечивает достаточного погружения для того, чтобы пользователь увидел реальную среду, так как поле зрения ограничено размерами монитора. В продаже имеются шторные очки *Crystal Eyes* от *Stereographics* и *SGS* от *Tektronix*.

Устройства осязания. Устройство осязания дает ощущение физического прикосновения. Такое устройство позволяет пользователям почувствовать реальный объект через систему силовой обратной связи, создающую иллюзию работы с реальным материалом. Примером устройства осязания является джойстик с силовой обратной связью, через который пользователь ощущает силу реакции на рабочей руке. В продаже имеются джойстики *Haptic Mater* от *Nissho Electronics* и *BSP Joystick* от *AEA Technologies*.

Прибор *Phantom* фирмы *SensAble Technology* – это устройство осязания на базе карандаша, имеющее шесть степеней свободы. Более передовым типом устройства осязания является экзоскелет – сложная система механических звеньев, окружающая всю руку так, что каждый палец и сустав независимо получают силовую обратную связь. Он объединяет в себе информационную перчатку и устройство осязания. Примерами экзоскелетов являются *The Exoskeletal Hand Master* от *Sarcos* и *Force ArmMaster* от *Exos*.

Следящие системы. Следящие устройства используют электромагнитную, ультразвуковую, оптическую или механическую систему для определения положения и ориентации отслеживаемого объекта. Следящее устройство может быть встроено в головной дисплей для отслеживания направления взгляда и положения головы или в информационную перчатку – для отслеживания положения руки. Следящее устройство можно также прикрепить на любую часть тела. В продаже имеются такие следящие системы, как *Flock of Birds* от *Ascension* и *ISOTRAK II* от *Polhemus* (<http://www.polhemus.com>).

Типичным примером следящих устройств с использованием ультразвуковой технологии могут служить *Head-tracker* от *Logitech* (<http://www.logitech.com>) и *GAMS* от *Transition State Corporation*. Механическая система используется в устройствах *GyroEngine* от *Gyrations*, *ADL-1* от *Shooting Star Technology* и *Wrighttrack* от *Vidtronic*s. Наконец, к устройствам с использованием оптической технологии относятся *GRD-1010* от *GEC Ferranti*, *DynaSight* от *Origin Instruments* и *RtPM* от *Spatial Positioning Systems*.

Информационные перчатки. Информационная перчатка имеет на каждом суставе пальцев руки датчики, измеряющие изгиб пальца. Положение руки в целом определяется следящей системой, прикрепленной к перчатке. Обычно информация, получаемая с перчатки, преобразуется в виртуальной среде снова в изображение, форма и положение которого динамически изменяются, следуя за движениями руки пользователя. В продаже имеются информационные перчатки *Dexterous Hand master* от *Exos*, *CyberGlove GG1801* от *Virex* и *DataGlove* от *Greenleaf Medical Systems*.

Примеры промышленного применения виртуальной инженерии

Самолет Boeing 777. Boeing 777 – это первый коммерческий самолет, успешно спроектированный безбумажным методом. Для разработки модели 777 корпорация *Boeing* организовала 238 межфункциональных групп проектирования и изготовления,

ответственных за конкретные продукты. Компания *Boeing* использовала CAD-систему *CATIA* от *Dassault/IBM* и разработала собственную систему предварительной компьютерной сборки *EPIC (Electronic Preassembly in Computer)*. Перед сборкой первого самолета не было изготовлено ни одного физического прототипа, кроме макета носовой части (для проверки критической проводки). Виртуальное прототипирование позволило компании *Boeing* вовлечь в процесс проектирования самолета заказчиков и операторов (до линейных механиков). Виртуальное прототипирование было настолько успешным, что несоосность при монтаже левого крыла составила всего 0,03 мм.

Виртуальный прототип локомотивного двигателя фирмы GM. Подразделение *Electro-Motive* фирмы *General Motors* разработало локомотивный двигатель *GM16V265H* мощностью 6300 л.с. путем так называемой «виртуальной разработки продукта» в сотрудничестве с *Unigraphics Solutions*. Все детали были представлены в виде трехмерных моделей, и последующий анализ, оптимизация конструкции и программирование станков с ЧПУ и приспособлений проводились для этих моделей. Моделирование позволило обеспечить поставку продукта в сжатые сроки: первый двигатель был построен через 18 месяцев после начала программы, в то время как обычно подобный процесс занимает более 36 месяцев. Это позволило также гораздо быстрее провести тесты надежности, что дало GM возможность удовлетворить более жесткие требования к надежности.

Дизайн интерьера салона автомобиля фирмы Chrysler. Дизайн интерьера салона автомобиля *Dodge Durango* 1998 г. был разработан методом виртуального проектирования. Член дизайнерской группы сидит в упрощенном макете салона автомобиля, состоящем только из сиденья, руля и педалей. Дизайнер, на котором надеты головной дисплей системы виртуальной реальности, информационные перчатки и датчики движений, рассматривает виртуальный прототип интерьера автомобиля (приборная доска, органы управления радиоприемника, бардачок и окна) и взаимодействует с ним. Виртуальный прототип позволяет легко вносить изменения в дизайн и оценивать обзор, доступность и эстетику.

Этот метод дает возможность быстро проверять различные варианты дизайна.

Поезд Metrocar 2000 в Стокгольме. Metrocar 2000 – это новая система общественного транспорта в Стокгольме, разработанная компанией *Adtranz Sweden*. Используя программу *dVISE* от *Division*, *Adtranz* сконструировала виртуальные прототипы поездов, включая полностью оснащенные интерьеры с текстурными сидениями, полами, рекламными плакатами и индикаторной панелью машиниста. Глядя на виртуальный прототип, зрители могут получить представление о масштабе, пространственных отношениях и эстетике дизайна. Это позволяет заказчику и инженеру на ранних стадиях проектирования знакомиться с дизайном продукта и вносить в него изменения.

Проектирование кораблей для Королевского военно-морского флота Великобритании. Морской директорат перспективных проектов Великобритании внедрил виртуальное проектирование в рамках своей новой программы разработки кораблей. С помощью программы *ENVISION* от *Deneb Robotics* методом имитации в интерактивном режиме был разработан кварталдек для нового корабля. Имитация включала движение корабля (с шестью степенями свободы) и его влияние на работающее оборудование, людей на борту и условия освещения в каютах. Это моделирование позволило инженерам оценить проект и внести в него необходимые изменения.

В заключение отметим, что виртуальная инженерия – зарождающаяся технология. Она обладает достаточным потенциалом для того, чтобы стать значительной составляющей деятельности инженера, однако на сегодняшний день функциональность и возможности применения систем виртуальной инженерии ограничены. Чтобы виртуальная инженерия превратилась в развитую технологию, необходимо получить возможность полностью отразить функциональное поведение физических систем посредством компьютерной имитации. Обсудим некоторые связанные с этим проблемы.

Новые средства проектирования. Виртуальное проектирование представляет принципиально новую среду для разработки.

В ней зрение является стереоскопическим, а взаимодействие с моделью конструкции осуществляется с помощью нескольких органов чувств. Эта новая среда открывает возможности для появления новых методов проектирования и подходов к моделированию. В ближайшем будущем конструктор будет иметь возможность взять объект в руки и растянуть его или создать и изменить модель с помощью одного только голоса. Новый подход к моделированию обеспечит более естественные и интуитивные способы создания моделей.

Моделирование процессов и физических объектов. В настоящее время возможности имитации сводятся главным образом к кинематике. Моделирование динамических, деформируемых и жидких систем обычно требует анализа методом конечных элементов, отнимающего большое количество вычислительных ресурсов. Чтобы это имело какую-то ценность в качестве средства виртуального проектирования, данный анализ необходимо производить в реальном времени, а чтобы стала возможной имитация в реальном времени, необходимо компактное и точное моделирование. Более того, модели должны содержать в себе информацию о своих физических свойствах и экспериментальные данные, демонстрирующие их физическое поведение.

Мера возможности производства. Производственные процессы различны, каждый имеет свои собственные уникальные характеристики. Таким образом, найти какую-то общую методику, определяющую возможности производства различных продуктов, представляется трудной задачей. Необходимы исследования определений возможности производства и методологии ее оценки. Помимо простого решения типа «да/нет» необходимо определить количественную меру возможности производства. Кроме того, оценку возможности производства необходимо трансформировать в оценку продолжительности производственных процессов и затрат.

Быстродействие системы. В настоящий момент качество визуализации и имитации сильно ограничивается недостаточным быстродействием системы. Благодаря экспоненциальному росту скорости обработки и прогрессу технологии распределенных

вычислений качество имитации улучшается. Однако быстродействие системы остается все еще слишком низким для полноценного виртуального проектирования. Для визуализации требуется детализированное трехмерное отображение и анимация с высокой частотой кадров. Оценка проекта включает в себя анализ динамических систем и оптимизации, требующих больших вычислительных ресурсов. Кроме того, для обеспечения коллективной разработки необходимо повысить скорость работы сетей, расширить полосу пропускания и увеличить число каналов.

Стандарт интерфейса данных. Виртуальное проектирование включает в себя взаимодействие различных пакетов прикладных программ. Моделирование детали обычно производится в САД-системе, анализ – в программе анализа по методу конечных элементов, а компьютерная имитация – в интерактивной системе имитации производства. Для коллективной разработки необходимо, чтобы эти различные системы работали вместе. Стандартные интерфейсы баз данных и программного обеспечения являются ключом к виртуальному проектированию.

Открытая архитектура. Открытая архитектура придает системе масштабируемость. Системы виртуального проектирования необходимо объединять с имеющимися в настоящий момент инженерными системами для получения дополнительной функциональности или решения разнотипных задач. Открытая архитектура позволяет системе задействовать большой резерв инженерных ресурсов и находить разнообразные формы применения.

4.2. Структура и разновидности систем автоматизированного проектирования

Как и любая сложная система, САПР состоит из подсистем. Различают подсистемы проектирующие и обслуживающие.

Проектирующие подсистемы непосредственно выполняют проектные процедуры. Примерами проектирующих подсистем могут служить подсистемы геометрического трехмерного

моделирования механических объектов, изготовления конструкторской документации, схемотехнического анализа, трассировки соединений в печатных платах.

Обслуживающие подсистемы обеспечивают функционирование проектирующих подсистем, их совокупность часто называют системной средой (или оболочкой) САПР. Типичными обслуживающими подсистемами являются подсистемы управления проектными данными, подсистемы разработки и сопровождения программного обеспечения *CASE (Computer Aided Software Engineering)*, обучающие подсистемы для освоения пользователями технологий, реализованных в САПР.

Структурирование САПР по различным аспектам обуславливает появление *видов обеспечения САПР*. Принято выделять семь видов обеспечения САПР:

- *техническое (ТО)*, включающее различные аппаратные средства (ЭВМ, периферийные устройства, сетевое коммутационное оборудование, линии связи, измерительные средства);

- *математическое (МО)*, объединяющее математические методы, модели и алгоритмы для выполнения проектирования;

- *программное*, представляемое компьютерными программами САПР;

- *информационное*, состоящее из базы данных, СУБД, а также включающее другие данные, используемые при проектировании (отметим, что вся совокупность используемых при проектировании данных называется информационным фондом САПР, а база данных вместе с СУБД носит название банка данных);

- *лингвистическое*, выражаемое языками общения между проектировщиками и ЭВМ, языками программирования и языками обмена данными между техническими средствами САПР;

- *методическое*, включающее различные методики проектирования; иногда к нему относят также математическое обеспечение;

- *организационное*, представляемое штатными расписаниями, должностными инструкциями и другими документами, регламентирующими работу проектного предприятия.

Классификацию САПР осуществляют по ряду признаков, например по приложению, целевому назначению, масштабам

(комплексности решаемых задач), характеру базовой подсистемы – ядра САПР.

По приложениям наиболее представительными и широко используемыми являются следующие группы САПР:

1. САПР для применения в отраслях общего машиностроения. Их часто называют машиностроительными САПР или системами *MCAD (Mechanical CAD)*.

2. САПР для радиоэлектроники: системы *ECAD (Electronic CAD)* или *EDA (Electronic Design Automation)*.

3. САПР в области архитектуры и строительства.

Кроме того, известно большое число специализированных САПР или выделяемых в указанных группах, или представляющих самостоятельную ветвь в классификации. Примерами таких систем являются САПР больших интегральных схем (БИС), САПР летательных аппаратов, САПР электрических машин и т. п.

По целевому назначению различают САПР, обеспечивающие разные аспекты (страты) проектирования. Так, в составе *MCAD* появляются рассмотренные выше *CAE/CAD/CAM*-системы.

По масштабам различают отдельные программно-методические комплексы (ПМК) САПР, например: комплекс анализа прочности механических изделий в соответствии с методом конечных элементов (МКЭ) или комплекс анализа электронных схем; системы ПМК; системы с уникальными архитектурами не только программного (*software*), но и технического (*hardware*) обеспечения.

По характеру базовой подсистемы различают следующие разновидности САПР:

1. САПР на базе подсистемы машинной графики и геометрического моделирования. Эти САПР ориентированы на приложения, где основной процедурой проектирования является конструирование, т. е. определение пространственных форм и взаимного расположения объектов. К этой группе систем относится большинство САПР в области машиностроения, построенных на базе графических ядер.

В настоящее время широко используются унифицированные графические ядра, применяемые более чем в одной САПР (ядра *Parasolid* фирмы *EDS Unigraphics* и *ACIS* фирмы *Integrgraph*).

2. САПР на базе СУБД. Они ориентированы на приложения, в которых при сравнительно несложных математических расчетах прорабатывается большой объем данных. Такие САПР преимущественно встречаются в технико-экономических приложениях, например при проектировании бизнес-планов, но они имеются также при проектировании объектов, подобных щитам управления в системах автоматики.

3. САПР на базе конкретного прикладного пакета. Фактически это автономно используемые ПМК, например имитационного моделирования производственных процессов, расчета прочности по МКЭ, синтеза и анализа систем автоматического управления и т.п. Часто такие САПР относятся к системам САЕ. Примерами могут служить программы логического проектирования на базе языка VHDL, математические пакеты типа MathCAD.

4. Комплексные (интегрированные) САПР, состоящие из совокупности подсистем предыдущих видов. Характерными примерами комплексных САПР являются САЕ/CAD/CAM-системы в машиностроении или САПР БИС. Так, САПР БИС включает в себя СУБД и подсистемы проектирования компонентов, принципиальных, логических и функциональных схем, топологии кристаллов, тестов для проверки годности изделий. Для управления столь сложными системами применяются специализированные *системные среды*.

4.3. Программные системы проектирования (примеры программ) [3]

К числу мировых лидеров в области CAD/CAM/CAE-систем верхнего уровня относятся системы *Unigraphics* (компания *EDS*), *CATIA* (*Dessault Systems*), *Pro/Engineer* (*PTC*). Продолжают использоваться также системы *I-DEAS* (*EDS*), *CADDS5* (*PTC*) и *EUCLID3* (*Matra Datavision*).

Вначале рассмотрим структуру ПО САПР и его функциональные возможности на примере комплекса программ *Pro/Engineer*.

Комплекс насчитывает несколько десятков программ (модулей), которые подразделены на группы программ конструкторского проектирования механических объектов, промышленного дизайна, функционального моделирования, технологического проектирования, обмена данными.

Базовые модули конструкторского проектирования (подсистема CAD) предназначены для твердотельного и поверхностного моделирования, синтеза конструкций из базовых элементов формы (БЭФ), поддержки параметризации и ассоциативности, проекционного черчения и разработки чертежей с простановкой размеров и допусков. Пользователь может пополнять библиотеку БЭФ оригинальными моделями. Синтез трехмерных модулей сложной формы возможен вытягиванием плоского контура по нормали к его плоскости, его протягиванием вдоль произвольной пространственной кривой, вращением контура вокруг заданной оси, натягиванием между несколькими заданными сечениями. Синтез сборок выполняется вызовом или ссылкой на библиотечные элементы, их модификацией, разработкой новых деталей. Детали сборки можно нужным образом ориентировать в пространстве. Далее следует ввести ассоциативные (сопрягающие) связи. Дополнительные модели конструкторского проектирования имеют более конкретную, но узкую специализацию. Примерами таких модулей могут служить модули конструирования панелей из композиционных материалов, разработки штампов и литейных пресс-форм, трубопроводных систем, сварных конструкций, разводки электрических кабелей и жгутов.

Модули функционального моделирования (подсистема CAE) используются как препроцессоры и постпроцессоры в программах конечно-элементного анализа (нанесение сетки конечных элементов, визуализация результатов анализа), для анализа теплового состояния конструкций, оценки виброустойчивости и др.

Основные модули технологического проектирования (подсистема CAM) служат для моделирования технологических процессов фрезерной, токарной, электроэрозионной обработки и для разработки постпроцессоров для систем управления оборудованием с ЧПУ.

Модули обмена данными (конверторы форматов данных) должны обеспечивать возможности импорта и экспорта данных в другие CAE/CAD/CAM-системы.

Система Unigraphics – универсальная система геометрического моделирования и конструкторско-технологического проектирования, в том числе разработки больших сборок, прочностных расчетов и подготовки конструкторской документации. Система многомодульная.

В конструкторской части (подсистема CAD) имеются средства для твердотельного конструирования, геометрического моделирования на основе сплайновых моделей поверхностей, создания чертежей по 3D-модели, проектирования сборок (в том числе с сотнями и тысячами компонентов) с учетом ассоциативности, анализа допусков и др.

В технологической части (подсистема CAM) предусмотрены разработка управляющих программ для токарной и электроэрозионной обработки, синтез и анализ траекторий инструмента при фрезерной трех- и пятикоординатной обработке, при проектировании пресс-форм, штампов и др. Для инженерного анализа (подсистема CAE) в систему включены модели прочностного анализа с использованием МКЭ с соответствующими пре- и постпроцессорами, кинематического и динамического анализа механизмов с определением сил, скоростей и ускорений, анализа процессов литья пластических масс.

Другая система верхнего уровня CATIA позволяет заказчику генерировать собственный вариант САПР сквозного проектирования – от создания концепции изделия до технологической поддержки производства и планирования производственных ресурсов. В системе реализовано поверхностное и твердотельное 3D-моделирование и оптимизация характеристик изделий. Возможны фотореалистичная визуализация, восстановление математической модели из материального макета. Система масштабируема. Предлагаются типовые конфигурации, в том числе варианты для полнофункционального сквозного проектирования сложных изделий и проектирования комплектующих на небольших и средних предприятиях. Аналогичные возможности реализованы и в других тяжелых САПР.

Значительно дешевле обходится приобретение САПР среднего уровня. В России получили распространение системы компаний *Autodesk*, *Solid Works Corporation*, *Топ Системы*, *Аскон*, *Интермех*, *Bee-Pitron* и некоторых других. Все эти системы ориентированы в первую очередь на платформу *Wintel*, как правило, имеют подсистемы: конструкторско-чертежную 2D, твердотельного 3D-моделирования, технологического проектирования, управления проектными данными, ряд подсистем инженерного анализа и расчета отдельных видов машиностроительных изделий, а также библиотеки типовых конструктивных решений. Широкое распространение в России и за рубежом получило ПО машиностроительных САПР компании *Autodesk*.

Линия современных программных систем конструкторско-го проектирования фирмы *Autodesk* включает ряд систем, среди которых наиболее развитыми следует считать системы *AutoCAD*, *Mechanical Desktop* и *Inventor*.

Система *Mechanical Desktop (MDT)* предназначена для параметрического 3D-моделирования, ассоциативного конструирования, распределенного проектирования в сети Интернет, оформления 2D-документации. Построена на графическом ядре *ACIS*. *Имеется управляющая программа САД-менеджер со средствами настройки, конфигурирования и управления рабочими группами.*

Система *Inventor* предназначена для твердотельного параметрического проектирования, ориентирована на разработку больших сборок с сотнями и тысячами деталей, имеет развитую библиотеку стандартных элементов. В основе системы также лежит графическое ядро *ACIS*. Построение 3D-моделей возможно выдавливанием, вращением, по сечениям, по траекториям. Из 3D-модели можно получить 2D-чертежи и спецификации материалов. Поддерживается коллективная работа над проектом, в том числе в пределах одной и той же сборки. Предусмотрена автоматическая проверка кинематики, размеров детали с учетом положения соседних деталей в сборке. *Значительные удобства работы конструкторов обусловлены тем, что ассоциативные связи задаются не путем описания операций с параметрами и уравнений, а непосредственно определением формы и положения компонентов.*

В число продуктов *Autodesk* входит ряд других программ автоматизированного проектирования, в том числе *Autodesk Data Exchange* – набор конверторов для взаимного преобразования данных из форматов *DXF* и *SAT* (формат ядра *ACIS*) в такие форматы, как *STEP*, *IGES*, *VDA-FS*.

Ряд продуктов, интегрированных с программами проектирования компании *Autodesk*, создан компаниями, входящими в ассоциацию *Mechanical Applications Initiative* производителей прикладного ПО. Среди них следует отметить программу *Dynamic Designer Motion* (компания *Mechanical Dynamic*), предназначенную для расчетов динамики и кинематики механизмов (в том числе трехмерных). Элементами являются модели шарниров, пружин, сухого трения, ударных нагрузок.

Программа *Dynamic Designer Motion* имеет связи с группой программ конечно-элементного анализа *Cosmos*. Например, можно использовать программу *Cosmos/DesignSTAR* как автономно, так и в связке с программами *Inventor Solid Edge*, а программу *Cosmos/Works* – с программой *Solid Works*. С помощью этих программ проводят анализ деформированного состояния деталей, стационарных и нестационарных тепловых процессов, динамики жидкостей и газов, низкочастотных электромагнитных полей, определяют собственные частоты колебания конструкций.

Система твердотельного параметрического моделирования механических конструкций *Solid Works* (компания *Solid Works Corporation*) построена на графическом ядре *Parasolid*, разработанном в *Unigraphics Solution*. Синтез конструкции начинается с построения опорного тела с помощью операций типа выдавливания, протягивания или вращения контура с последующим добавлением и (или) вычитанием тех или иных тел. Используется технология граничного моделирования (*B-representation*) с аналитическим или сплайновым описанием поверхностей. При проектировании сборок на основе БЭФ можно задавать различные условия взаимного расположения деталей, автоматически контролировать зазоры и отсутствие взаимопересечения деталей. Предусмотрены *IGES*, *DXF*, *DWG*-интерфейсы с другими системами.

Среди САПР среднего уровня наряду с продуктами зарубежных фирм неплохо зарекомендовали себя системы отечественных разработчиков – это, прежде всего, системы *Компас* (компания Аскон) и *T-Flex CAD (Топ Системы)*.

В системе *Компас* для трехмерного твердотельного моделирования используется оригинальное графическое ядро. Синтез конструкций выполняется с помощью булевых операций над объемными примитивами, модели деталей формируются путем выдавливания или вращения контуров, построением по заданным сечениям. Возможно задание зависимостей между параметрами конструкции, расчет масс-инерционных характеристик. Разработка проектно-конструкторской документации, в том числе различных спецификаций, выполняется подсистемой *Компас-График*.

Имеются библиотеки с данными о типовых деталях и с графическими изображениями, а также программы специального назначения (проектирование тел вращения, пружин, металлоконструкций, трубопроводной арматуры, штамповой оснастки, выбора подшипников качения, раскроя листового материала и др.). Проектирование технологических процессов выполняется с помощью подсистемы *Компас-Автопроект*, программирование объемной обработки на станках с ЧПУ – с помощью подсистемы *ГЕММА-3D*. Ряд необходимых функций управления проектными данными возложено на подсистему *Компас-Менеджер*.

Подсистема трехмерного твердотельного моделирования *T-Flex CAD 3D* в САПР *T-Flex CAD* построена на базе ядра *Parasolid*. Реализована двунаправленная ассоциативность, т. е. изменение параметров чертежа автоматически вызывает изменение параметров модели, и наоборот. При проектировании сборок изменение размеров или положения одной детали ведет к корректировке положения других. Модель 3D может быть получена непосредственно по имеющемуся чертежу, или с помощью булевых операций, или путем выталкивания, протягивания, вращения профиля, лофтинга и т. п. Предусмотрен расчет масс-инерционных параметров.

В то же время можно по видам и разрезам трехмерной модели получить чертеж, для чего используется подсистема *T-Flex CAD*

3D SE. Для параметрического проектирования и оформления конструкторско-технологической документации служит подсистема *T-Flex CAD 2D*, для управления проектами и документооборотом – подсистема *T-Flex DOCs*. В подсистеме технологического проектирования *T-Flex/ТехноПро* выполняются синтез технологических процессов, расчет технологических размеров, выбор режущего и вспомогательного инструмента, формирование технологической документации, в том числе операционных и маршрутных технологических карт, ведомостей оснастки и материалов, карт контроля. Подготовка программ для станков с ЧПУ осуществляется в подсистеме *T-FlexЧПУ*. Кроме названных основных подсистем в состав *T-Flex CAD* включен ряд программ для инженерных расчетов деталей, проектирования штампов и пресс-форм.

В САПР *Cadmech 2000* белорусской компании *Интермех* входят программы *AVS* для выпуска конструкторской документации, *Techcard* для технологической подготовки производства, *LCAD* для планирования производственных цехов и участков и др. Для собственно конструкторского 3D-проектирования *Интермех* использует программы компании *Autodesk*.

Разработкой продуктов для САПР литейного производства занимается компания *Moldflow*, ее программы *Part Adviser* и *Mold Advisor* предназначены для моделирования процессов литья пластмасс.

Важное место в конструкторско-технологических САПР занимают программы технологической подготовки производства. Компания *Consistent Software* предлагает систему *TechnologiCS* для технологической подготовки дискретного производства. Эта система выполняет функции составления спецификаций, ведения дерева проекта и библиотеки чертежей, синтеза технологических процессов, выбора инструмента, расчета режимов резания, нормирования расхода материалов, ведения технологической документации. Система *SolidCAM (CADTech)*, построенная, как и *Mechanical Desktop*, на ядре *ACIS*, служит для получения управляющих программ для токарной, 2,5- и 3-осевой фрезерной обработки на станках с ЧПУ. Система *ТЕХТРАН*

(НИИ «Информатика») включает модули токарной, фрезерной, электроэрозионной обработки.

Мировыми лидерами среди программ конечно-элементного анализа являются программно-методические комплексы *Nastran* и *Patran* (компания *MSC Software Corporation*) и *Ansys* (компания *Ansys Inc.*). Как правило, эти комплексы включают в себя ряд программ, родственных по математическому обеспечению, интерфейсам, общности некоторых используемых модулей. Эти программы различаются ориентацией на разные приложения, степенью специализации, ценой или выполняемой обслуживающей функцией. Например, в комплексе *Ansys* основные решающие модули позволяют выполнять анализ механической прочности, теплопроводности, динамики жидкостей и газов, акустических и электромагнитных полей. Во все варианты программ входят пре- и постпроцессоры, а также интерфейс с базой данных. Предусмотрен экспорт (импорт) данных между *Ansys* и ведущими комплексами геометрического моделирования и машинной графики.

Мировой лидер среди средств моделирования механических процессов на макроуровне путем решения СОРУ – программа *Adams*, а примером отечественных систем подобного назначения следует назвать программы *ПА7* и *ПА9*. В САПР крупных предприятий обычно используют программы разных уровней. Связано это с тем, что более 80 % всех процедур конструирования можно выполнить на САД-системах нижнего и среднего уровней, кроме того, «тяжелые» системы дороги. Поэтому предприятие приобретает лишь ограниченное число экземпляров (лицензий) программы верхнего уровня, а большинство клиентских рабочих мест обеспечивается экземплярами программ нижнего или среднего уровней. При этом возникает проблема обмена информацией между разнотипными САД-системами. Она решается путем использования языков и форматов, принятых в *CALS*-технологиях, хотя для неискаженной передачи геометрических данных с помощью промежуточных унифицированных языков приходится преодолевать определенные трудности.

Контрольные вопросы

1. Дайте общую характеристику автоматизированным системам CAD/CAM/CAE.

2. Опишите функции CAD-систем (характеристики параметризации и ассоциативности, геометрическое моделирование).

3. Дайте характеристику основным функциям САМ-систем, включая автоматизацию технологической подготовки производства (САПП).

4. Опишите модифицированный и генеративный подходы к САПП.

5. Опишите групповую технологию проектирования и подготовки производства.

6. Обоснуйте необходимость физического моделирования в процессах проектирования (включая технологию быстрого прототипирования).

7. Дайте общую характеристику методам быстрого прототипирования и изготовления (БПИ).

8. Опишите методы стереолитографии, отверждения на твердом основании, избирательного лазерного спекания.

9. Опишите методы трехмерной печати, ламинирования, моделирования методом наплавки.

10. Дайте характеристику основным функциям CAE-систем и перспектив их развития применительно к разработке методов виртуальной инженерии.

11. Опишите структуру и функции виртуальной инженерии.

12. Укажите и опишите основные компоненты виртуальной инженерии.

13. Опишите основное оборудование виртуальной реальности (устройства ввода и вывода информации).

14. Опишите структуру и классификацию существующих САПП.

15. Дайте характеристику системам САПП верхнего уровня Unigraphics, CATIA, Pro/Engineer.

16. Дайте сравнительный анализ системам САПП среднего уровня.

Лекция 5

ИНТЕГРИРОВАННЫЙ ПОДХОД К РАЗРАБОТКЕ МЕХАТРОННЫХ СИСТЕМ

Наибольший успех при разработке сложных (мехатронных) систем может быть достигнут только в случае полной интеграции отдельных подсистем САПР. Поэтому в последние годы получили широкое распространение универсальные интегрированные CAD/CAM/CAE-системы, которые предназначены для комплексной автоматизации процессов проектирования, конструирования и производства различной продукции. Они входят в состав современных САПР. На практике применяется большое число интегрированных систем различных версий и конфигураций для решения конкретных задач проектирования. Такая ситуация порождает большие издержки проектирования, особенно если речь идет об участии большого количества исполнителей при проектировании сложных производственных систем.

Кроме того, в связи с большим объемом технических данных по мере разработки проекта возникает проблема переноса данных из одной системы в другую. Поэтому современные системы должны включать в свой состав программные пакеты, называемые системами управления данными в продуктах – PDM (product data management). Использование такой системы улучшает взаимодействие и повышает эффективность управления проектами. Универсальные CAD/CAM/CAE/PDM-системы обеспечивают следующие функции проектирования:

- трехмерное графическое моделирование;
- различные виды анализа и инженерных расчетов с использованием метода конечных элементов (МКЭ), моделирование кинематики и динамики механизмов и процессов;
- автоматизацию технологической подготовки производства;
- собственные или коммерческие СУБД;

- создание прикладных приложений концептуального проектирования и программного обеспечения сложных систем;
- управление данными проекта;
- обмен данными в различных графических форматах.

В качестве примера интегрированных САПР можно указать системы верхнего уровня: Catia, Unigraphics, Pro/Engineer, SolidWorks.

Далее рассмотрим программное обеспечение, позволяющее создавать и поддерживать интегрированные информационные среды для промышленных автоматизированных систем.

5.1. Системы PDM

Системы PDM предназначены преимущественно для информационного обеспечения проектирования – упорядочения информации о проекте, управление документами и документооборотом (включая спецификации и другие виды представления данных), обеспечения доступа к данным по различным атрибутам, навигации по иерархической структуре проекта. В ряде систем PDM поддерживают информационные связи не только внутри САПР, но и с производственной и маркетинговой документацией [1].

Если рассматривать ситуацию в динамике, то маркетинговые концепции должны передаваться в отдел планирования продуктов и в группу проектирования, которая взаимодействует с производственным отделом и группой поддержки. Сведения об обнаруженных проблемах возвращаются в отделы проектирования и производства. Сведения о продажах и проектах передаются поставщикам и партнерам. Данные о расходах обрабатывает бухгалтерия. Все эти виды деятельности тесно связаны друг с другом, поэтому изменения в содержании или состоянии проекта должны быть доступны всем участникам во избежание дорогостоящих ошибок.

Рассмотрим типичные для систем PDM основные функции [1, 3]:

1) поддержка интерактивной работы пользователя при создании моделей изделий (процессов), структурирование описаний

проектируемых объектов, предъявление пользователю этой иерархической структуры вместе с возможностями навигации по структуре (дереву изделия, проекта) и получение нужной информации по каждой указанной пользователем структурной компоненте;

2) управление конфигурацией изделия (процесса), ведение состава и контроль изменения изделия, сохранение истории проекта, классификация и кодификация изделий;

3) обеспечение целостности проектных данных при управлении версиями проекта и внесение изменений в проект (при изменении версии проекта сначала создается новая версия проекта, а изменения вносятся уже в эту новую версию; одна версия каждого объекта является текущей или активной версией; нельзя одновременно вносить изменения в один и тот же проект разным разработчикам);

4) управление документами и документооборотом, электронным архивом, поддержка классификаторов и справочников, автоматизированное составление спецификаций (в интегрированных САПР в документооборот входит большое число документов, связанных с процедурами маркетинга, планирования, администрирования и т. п.);

5) Использование полнофункциональных систем делопроизводства, разработанных независимо от конкретных PDM (ввод в оборот документов с помощью средств их автоматического распознавания и атрибутирования, поиск нужных данных, поддержка групповой работы над документами, разграничение прав доступа к документам, подготовка отчетов, маршрутизация документов, учет их движения, контроль исполнения предписанных документами действий, автоматическое уведомление соответствующих лиц, планирование работ, связанных с прохождением документов);

6) управление проектами (процессом проектирования) включает распределение основных и вспомогательных заданий и контроль их выполнения, выполнение проектных операций и процедур, пересылку документов и файлов конкретным пользователям, изменение статуса объекта, просмотр и утверждение инженерных проектов, внесение в них изменений и т. п.;

- 7) визуализация структуры изделия (проекта) в виде дерева, многооконное представление трехмерных изображений;
- 8) возможность подключения внешних систем;
- 9) использование криптографических средств и средств электронной подписи.

Интеграция данных на ранних этапах развития PDM связывалась только с организацией сквозного (параллельного) проектирования изделий и систем в рамках конкретной САПР. В настоящее время (как отмечалось выше) основным содержанием проблемы интеграции стало обеспечение интерфейса САПР с другими автоматизированными системами. Проблема решается с помощью поддержки типовых форматов, например путем конвертации данных из общепринятых форматов во внутренние представления конкретных САПР.

Примеры PDM

Первые известные PDM, созданные фирмами, разрабатывающими САПР или АСУП:

- iMAN (Unigrphics Solutions);
- Euclib Design Manager (Matra DataVision);
- Smar Team (Smart Solutions, Dessault Sysytems);
- Baan PDM (BAAN);
- Optegra и ProPDM (PTC);
- ENOVIA (IBM, Dessault Sysytems).

Дополнительные преимущества получены за счет совместного использования CATIA (Dessault Sysytems) и ENOVIA.

На роль PDM претендует система ведения архива технической документации и управления проектными данными Search (Интермех).

Другие примеры PDM:

- STEP Suite;
- Outdo CS PDM (Consistent Software);
- PartY Plus (Лоция Софт);
- Omega Production (СИКОР) может служить основой для интеграции САПР и АСУП.

5.2. Метод обмена данными

Когда две или более CAD/CAM/CAE-системы объединяются и связываются в единое приложение для совместного использования данных, часто возникает проблема обмена данными. Фактически всегда существует проблема (потребность) связать воедино несколько систем либо внутри одной организации, либо внешне, как в случае со смежниками (по проекту) или поставщиками компонентов. Для решения этой коммуникационной проблемы необходима возможность преобразовать данные технических требований одной системы в форму, понятную для других систем, и наоборот. Чтобы облегчить преобразование и не разрабатывать программы-конверторы для всех возможных пар САПР, было предложено несколько стандартных форматов для хранения данных технических требований. Вкратце рассмотрим типичные стандартные форматы.

Различные CAD/CAM/CAE-системы хранят данные технических требований в структурах разного вида, поэтому для переноса данных необходимо преобразовать данные технических требований одной системы в формат другой. Еще один конвертор (преобразователь) необходим для переноса данных между двумя системами в противоположном направлении. Следовательно, для каждой пары систем необходимо иметь два конвертора. Двухнаправленные стрелки для каждой пары систем (рис. 5.1, а) предполагают наличие двух конверторов. Эти конверторы для каждой конкретной пары систем называются *прямыми конверторами* (*direct translators*). Если у нас есть n различных систем, нам необходимо разработать $n(n - 1)$ конверторов, поскольку количество пар систем равно $n(n - 1)/2$. Например, для обмена данными между 10 системами придется разработать 90 конверторов. Таким образом, метод прямого конвертирования непрактичен, так как требует разработки слишком большого количества конверторов при необходимости работать с множеством систем. Более того, добавление одной системы к n уже имеющимся потребует написания $2n$ дополнительных конверторов.

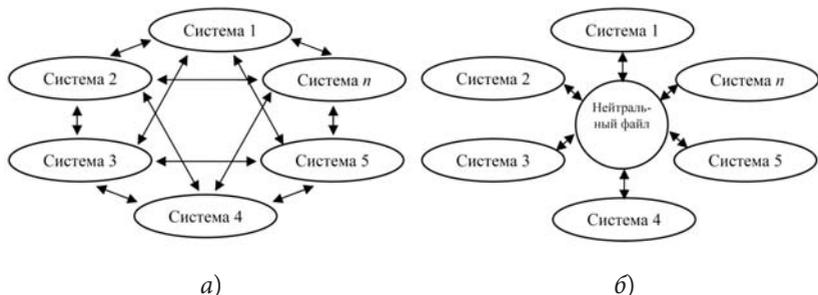


Рис. 5.1. Два метода обмена данными между двумя различными системами

Однако обмен данными можно обеспечить, введя нейтральную структуру базы данных, называемую *нейтральным файлом* (*neutral file*), которая была бы независима от существующих САПР. Эта структура будет действовать как промежуточная точка коммуникации между различными структурами баз данных САПР (рис. 5.1, б).

Таким образом, в каждой системе будет своя пара конверторов для экспорта и импорта данных в этот нейтральный формат. Конвертор, преобразующий данные из собственного формата данной системы в нейтральный формат, называется *препроцессором* (*pre-processor*), а конвертор, выполняющий обратное преобразование, – *постпроцессором* (*post-processor*) (рис. 5.2). Соответственно в этом случае для обмена данными между n системами потребуется $2n$ конверторов и лишь два дополнительных конвертора необходимо будет добавить при введении новой системы.

Иными словами, этот косвенный метод свободен от присущего прямому методу недостатка, когда требовалось писать всевозрастающее количество программ. Это основная причина, по которой косвенный метод принят в качестве главного метода обмена данными между различными системами, несмотря на то что в сравнении с прямым методом у него имеются некоторые недостатки. В частности, прямые конверторы работают быстрее косвенных, и создаваемые ими файлы данных обычно имеют

меньший размер, чем нейтральные файлы, генерируемые косвенными конверторами. Файл данных в собственном формате конкретной системы обычно также оказывается меньше нейтрального файла из-за обобщенной природы последнего. Когда мы переносим данные технических требований через нейтральный файл, некоторая информация, как правило, теряется, особенно информация о топологическом дереве и ограничениях в системах параметрического моделирования.

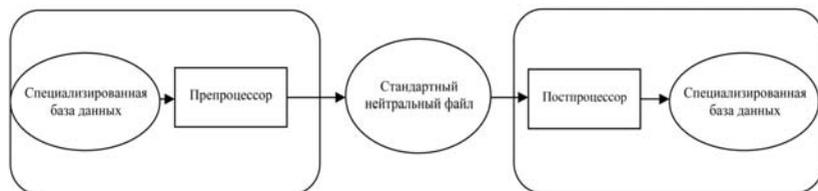


Рис. 5.2. Обмен данными с использованием нейтрального файла

Известны три типичных формата нейтрального файла: IGES (*Initial Graphics Exchange Specification* – первоначальная спецификация обмена графическими данными), DXF (*Drawing Interchange Format* – формат обмена чертежами) и STEP (*Standard for Exchange of Product model data* – стандарт обмена данными о модели продукта). В настоящее время IGES является самым популярным форматом нейтрального файла, а формат DXF используется главным образом для обмена данными чертежей. STEP – это стандартный формат данных, используемый для хранения полной информации обо всем жизненном цикле продукта, включая проектирование, анализ, производство, контроль качества, испытания и обслуживание помимо обычных данных технических требований. В настоящее время САД-системы, поддерживавшие формат IGES, ориентированы на переход к формату STEP.

В лекции 7 будут рассмотрены и другие способы организации информационных обменов.

5.3. Тенденции в развитии интегрированных систем [10, 11]

Сегодня информационная поддержка проектирования, конструирования и технологической подготовки сложных (мехатронных) производств, как правило, осуществляется с использованием преимущественно САПР верхнего уровня. Их выбор определяется способностью конкретных САПР решать поставленные перед производителем задачи.

Интегрированная CAD/CAM/CAE-система, дополненная средствами трехмерного проектирования и анализа, способными работать в распределенной компьютерной сети и удовлетворяющими принципам параллельного проектирования, определяет основную тенденцию ее развития.

Для дальнейшего развития интегрированных систем необходимо не только расширение функциональных возможностей системы проектирования, но и кардинального повышения производительности труда разработчиков, которая снижается в связи с возрастающим объемом обмена информацией между группами проектировщиков. Кроме того, требует решения еще одна проблема, связанная со сложностью переноса систем на другие платформы (например, с UNIX-платформы на платформу Wintel). Еще одна трудность возникает при необходимости ввести новый элемент в группу деталей, сохраненных в базе данных системы. Для этого соответствующая программная библиотека должна быть расширена, а затем перекомпилирована. Когда изменяются задачи проектирования, то проектировщики должны создавать новые прикладные подсистемы, не имея возможности применить разработанный ранее аппарат, аккумулирующий их «ноу-хау».

Поэтому предложенные в новых поколениях интегрированных систем их открытость и использование объектно-ориентированного проектирования (см. п. 3.3) направлены на преодоление некоторых указанных выше недостатков и ограничений.

Объектно-ориентированная модель определяется как электронная модель реального физического объекта, а не только как

представление его геометрии, и содержит не только геометрию объекта, но и все негеометрические его параметры, атрибуты и связи. Например, описание гидравлического пресса в виде «объекта» проектирования кроме размеров его конструктивных элементов будет содержать еще такие атрибуты, как «максимальное усилие», «величина давления в гидросистеме», «производительность» и «дата установки». В описание «объекта» также будут входить возможности его модификации, а также его отдельные компоненты, которые можно исследовать на прочность и надежность функционирования при различных эксплуатационных условиях.

Построенная на основе объектно-ориентированных моделей система CAD/CAM позволяет вводить описание объектов нового вида как подкласс более общего класса. Новый объект сможет наследовать не только атрибуты, но и методы преобразования уже известных объектов, что заодно существенно упрощает расширение базы данных конкретной системы и сокращает время проектирования объекта. В системы CAD/CAM/CAE закладываются возможности многократного повторного применения в разных проектах одних и тех же элементов конструкций и процессов.

Новая объектно-ориентированная архитектура открыта и обеспечивает широкое участие в совместном проектировании специалистов различных групп от заказчиков поставщиков комплекующих, до аналитиков и технологов.

Учитывая, что данные проекта – это самая ценная часть системы, то особенно тщательно должны быть разработаны все средства доступа к проектным данным и их обработки, обеспечивающие максимально высокую степень их непротиворечивости, защищенности и сохранности.

Возможность и направленность дальнейшего развития CAD/CAM/CAE-систем будут рассмотрены в лекции 7 «Информационная поддержка жизненного цикла мехатронных систем».

Контрольные вопросы

1. Дайте общую характеристику интегрированного подхода к разработке мехатронных систем.

2. Определите основное назначение систем PDM.
3. Опишите три первые (1–3) типичные функции PDM.
4. Опишите три следующие (4–6) типичные функции PDM.
5. Опишите три последние (7–9) типичные функции PDM.
6. Приведите примеры PDM.
7. Обоснуйте необходимость обмена данными между различными САПР.
8. Опишите метод прямого конвертирования обмена данными.
9. Опишите метод с использованием нейтрального файла обмена данными (косвенный метод).
10. Проведите сравнительный анализ двух методов обмена данными.
11. Опишите три типичных формата нейтрального файла.
12. Проведите сравнительный анализ трех типичных форматов нейтрального файла.
13. Опишите схему обмена данными с использованием нейтрального файла.
14. Перечислите дополнительные средства, необходимые для расширения функциональных возможностей CAD/CAM/CAE-систем.
15. Перечислите проблемы, которые необходимо решить для дальнейшего развития интегрированных CAD/CAM/CAE-систем.
16. Объяснить, почему открытость интегрированных систем и использование объектно-ориентированного проектирования направлены на дальнейшее развитие интегрированных систем.

Лекция 6

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ И МОДЕРНИЗАЦИЯ МЕХАТРОННЫХ СИСТЕМ

6.1. Основные положения

Следуя [12], отметим, что проектирование – сложная итерационная процедура, содержащая множество неформальных решений. И тем не менее существует определенная последовательность проектных действий.

Первым этапом (часто говорят верхним уровнем) проектирования является разработка концепции изделия (концептуальное проектирование). Концепция изделия (процесса) – это система взглядов на будущее изделие (производственного процесса).

Концептуальное проектирование выполняют в процессе предпроектных исследований (см. рис. 3.2), когда формируется техническое задание (ТЗ), технические предложения и эскизный проект будущего изделия (производственного процесса).

Все выше сказанное относится и к процедуре модернизации (реинжинирингу) деятельности предприятия, на котором реализуется производственный процесс.

Предпроектные исследования предприятий, на которых создаются или модернизируются производственные процессы, проходят путем обследования этих предприятий. На основе результатов обследования строят модель, отражающую деятельность предприятия на данный момент (до модернизации). Такую модель называют «AsIs’ (как есть). В процессе проектирования нового производственного процесса в качестве модели «AsIs’ принимается ее ближайший прототип.

Далее путем изменения модели «AsIs» разрабатывают новую модель производственного процесса, которую называют моделью «ToBe» (как должно быть). Модель «ToBe» – это сложная модель, которая имеет несколько аспектов описания. В ее состав входят *функциональные, информационные, поведенческие и структурные* модели.

Функциональная модель системы описывает совокупность выполняемых системой функций. Она должна отвечать на следующие вопросы [12]: «Какие функции выполняет система?», «Какова структура функций?», «Что есть вход и выход функций?», «Что обеспечивает управление функцией?», «Как связаны функции и компоненты системы?», «Как избежать дублирования функций?». В функциональную структуру должны войти, как минимум, все функции, приведенные в ТЗ на систему.

Информационная модель отражает структуры данных – их состав и взаимосвязи.

Поведенческая модель описывает функционирование системы (процессов) как упорядоченную последовательность событий с одновременным описанием объектов, имеющих непосредственное отношение к процессу. В ней фигурируют такие категории, как состояние системы, событие, переход из одного состояния в другое, условия перехода, последовательность событий. Если функциональная модель позволяет ответить на вопрос: «Что делает система?», то поведенческая модель объясняет «Как система это делает». В основе поведенческого моделирования лежат модели имитационного моделирования систем массового обслуживания, сети Петри, методы конечного автомата.

Структурная модель характеризует совокупность элементов системы и их взаимосвязь, описывает морфологию (построение) системы.

Концептуально современный подход к созданию и информационному обеспечению сложных (мехатронных) производственных процессов и систем базируется на следующих общих положениях:

1. Основой разработки мехатронных систем служат методы *совмещенного (параллельного)* проектирования, предполагающие

совмещение во времени некоторых этапов проектирования, выполняемых последовательно при традиционном проектировании (например, последовательно выполняется разработка механической, электронной, сенсорной и компьютерной частей системы с последующей разработкой интерфейсных блоков).

2. Синергетический характер мехатронных систем проявляется в том, что составные части системы не просто дополняют друг друга, но объединяются таким образом, что образованная ими система начинает обладать новыми свойствами. При этом мехатронные системы, в отличие от традиционных, обладают меньшей структурной избыточностью и большей степенью интеграции. В результате повышается конструктивная компактность системы (вплоть до миниатюризации в микромашинах), улучшаются массогабаритные и динамические характеристики машин, упрощаются кинематические цепи. Возникают дополнительные возможности при проектировании мехатронных систем, наилучшим образом отвечающие требованиям заказчика. Функциональное, структурное и конструктивное объединение разных по физической природе элементов, связанных между собой аппаратными и программными интерфейсами – основа мехатронного подхода к процессу проектирования производственных процессов и систем.

3. Интегрированные мехатронные элементы выбираются разработчиком уже на стадии проектирования машины, а затем обеспечивается необходимая инженерная и технологическая поддержка при производстве и эксплуатации машины. В этом радикальное отличие мехатронных машин от традиционных, когда зачастую пользователь был вынужден самостоятельно объединять в систему разнородные механические, электронные и информационно-управляющие устройства различных изготовителей. Именно поэтому многие сложные комплексы (например, некоторые гибкие производственные системы в отечественном машиностроении) показали на практике низкую надежность и невысокую технико-экономическую эффективность.

4. Проектирование мехатронных систем базируется на идее модульного конструирования, при котором из отдельных

многофункциональных модулей (механической, электронной, компьютерной частей системы) komponуются гибкие сложные системы модульной архитектуры.

5. В мехатронных системах наблюдается перераспределение функциональной нагрузки от аппаратных (например, электромеханических) модулей к информационным (компьютерным) модулям. При этом исключается многоступенчатое преобразование энергии и информации.

6. При проектировании мехатронных систем широко применяются методы *визуализации динамических процессов*, протекающих в системах, вплоть до использования систем *виртуальной реальности* при проектировании сложных технических систем.

7. Высокий уровень интеллектуализации мехатронных систем диктует необходимость проектирования интеллектуальных систем управления (для борьбы с неопределенностями, сопутствующими функционированию сложных динамических систем). При этом необходим обоснованный выбор интеллектуальных технологий при проектировании многоуровневых иерархических систем управления.

8. Использование концепции АКУ (аппарат конфигурируемого управления), у которой маневренность является приоритетной функцией. Ранее при создании машины сначала выполняли базовое проектирование (с точки зрения физики и механики), а затем в качестве подсистемы разрабатывали систему управления. Но оказалось, что можно спроектировать более эффективную машину, если еще на этапе базового проектирования заложить основы системы управления, используя принципы АКУ. При этом машина может реализовать свои функции только благодаря существованию системы управления. Таким способом можно добиться максимальной динамичности системы (машины). Устойчивость достигается благодаря наличию контуров управления. Концепция АКУ широко используется в самолетостроении.

9. Широкое использование «встроенных» систем (внедрение цифровых электронных блоков и микропроцессоров в мехатронные узлы машин и систем).

10. Бессенсорное управление – перенос функциональной нагрузки на интеллектуальные устройства мехатронных систем.

Практическое воплощение данных принципов требует привлечения всех современных методов проектирования и поддержки функционирования сложных (мехатронных) систем.

6.2. Инструментальные средства концептуального проектирования и модернизации систем

В современных информационных технологиях важное место отводится инструментальным средствам и средам разработки и модернизации мехатронных систем (производственных процессов и систем). Эти технологии и среды образуют системы, называемые CASE-технологиями (или CASE-системами).

CASE-технологии [3, 11]

Аббревиатура CASE имеет двойное толкование, соответствующее двум направлениям использования CASE-технологий. Первое из них Computer Aided Software Engineering, что переводится как «автоматизированное проектирование программного обеспечения». Соответствующие CASE-технологии называют инструментальными средами разработки программного обеспечения (ПО). В этом случае CASE-технологии часто отождествляют с инструментальными средами разработки ПО, называемыми средами быстрой разработки приложений (RAD – Rapid Application Development). Примерами широко известных инструментальных сред RAD являются Delphi, Borland, C++, PowerSoft, СУБД (Oracle). Кроме того, используются специализированные средства (Tools). Эти текстовые и графические редакторы, электронные таблицы, User-пакеты и языки для создания приложений в составе CAD/CAM-систем и т. п. Наряду с самостоятельными

RAD-системами имеются и RAD-системы в составе САПР. Это прежде всего системы CAS.CADE фирмы Matra Datavision.

Второе направление, в котором используются CASE-технологии, – это поддержка концептуального проектирования сложных, слабоструктурированных систем – Computer Aided System Engineering. Далее CASE-технологии этого направления будем называть *системами CASE для концептуального проектирования*.

Среди систем CASE для концептуального проектирования различаются, описанные выше системы *функционального, информационного и поведенческого* проектирования. Наиболее известной методикой функционального проектирования сложных систем является методика SADT (Structured Analysis and Design Technique), предложенная в 1973 г. Р. Россом и впоследствии ставшая основой стандарта IDEF0 (Integrated DEFinition 0) – язык функционального проектирования. Для информационного моделирования использована методика инфологического (информационного) проектирования баз данных IDEF1X. Для поведенческого (динамического) моделирования используется методика IDEF3.

Процесс построения функциональных, информационных и поведенческих моделей автоматизируется за счет применения поддерживающих методологию IDEF-систем – BPWin, ERWin, OOWin, Design/IDEF и др. При этом в начале методом CASE-технологий строится модель процесса (As Is) – «как есть», а затем, после ее анализа и оптимизации с учетом использования современных интегрированных CAD/CAM/CAE/PDM-систем и вновь предложенных компонентов, – модель (To Be) – «как нужно».

Подсистему CASE в составе системной среды САПР можно рассматривать как специализированную САПР, в которой объектом проектирования являются новые версии подсистем САПР, в частности версии, адаптированные к требованиям конкретного заказчика.

Методика IDEF0, IDEF1X, IDEF3 [1, 3]

Рассмотрим кратко методологию SADT. Подход SADT относится к классу формальных методов, используемых при анализе

и разработке модели системы. SADT-модель предназначена для полного, точного и адекватного описания системы, имеющей конкретное назначение. Это назначение, называемое целью модели, вытекает из формального определения модели в SADT: «*М моделирует А, если М отвечает на вопросы относительно А*». Разработку SADT-модели начинают с формулировки вопросов, на которые модель должна дать ответы, т.е. формулируется цель моделирования. Далее дается точное определение границ моделируемой системы (модель точно устанавливает, что является и что не является субъектом моделирования). Ограничивая субъект, SADT-модель помогает сконцентрировать внимание на описываемой системе и позволяет избежать включения посторонних субъектов. Таким образом, SADT-модель должна иметь единственный субъект. Кроме того, SADT требует, чтобы модель рассматривалась все время с одной той же позиции. Эта позиция называется «точкой зрения» данной модели и однозначно определяет цель и субъект, например, конкретного технологического (производственного) процесса. Иногда только одна из множества возможных точек зрения может дать описание, удовлетворяющее цели модели. Например, для создания согласованной модели производственного процесса нужно встать на точку зрения главного инженера предприятия.

После того как определены субъект, цель и точка зрения модели, начинается первая интеграция процесса моделирования по методике SADT. Субъект определяет, что включить в модель, а что исключить из нее. Точка зрения диктует автору модели выбор нужной информации о субъекте и форму ее подачи. Цель становится критерием окончания моделирования. Конечным результатом этого процесса является набор тщательно взаимосвязанных описаний, начиная с описания самого верхнего уровня всей системы и заканчивая подробным описанием технологических операций производственного процесса.

Каждое из таких взаимосвязанных описаний называется диаграммой. Модель SADT можно представить в виде древовидной структуры диаграмм, где верхняя диаграмма является наиболее общей, а самые нижние наиболее детализированы. Ниже будут приведены примеры таких диаграмм.

Министерство обороны США (DoD) использовало методику SADT в качестве части своей программы ICAM – Integrated Computer Aided Manufacturing (интеграция компьютерных и промышленных технологий) и назвала ее IDEF0 (Integrated DEfinition 0) – язык функционального моделирования.

IDEF0 – это более строгая реализация ранее предложенной методики SADT (*Structured Analysis and Design Technique*). Начиная с момента создания первой версии, методика успешно применялась для проектирования телефонных сетей, систем управления воздушными перевозками, производственных предприятий и др.

Описание объектов и процессов в IDEF0 выполняется в виде совокупности взаимосвязанных блоков (рис. 6.1), называемых блоками ICOM (*Input – Control – Output – Mechanism*), где *I* – вход, *C* – управление, *O* – выход, *M* – механизм.

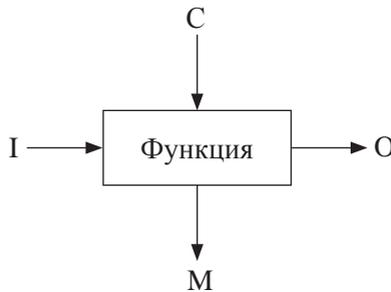


Рис. 6.1. Блок ICOM

Блоки представляют функции (работы), их названия выражаются глаголами или отглагольными существительными. Типичные примеры функций: планировать, разработать, классифицировать, измерить, изготовить, отредактировать, рассчитать, продать (или планирование, разработка, классификация, измерение, изготовление, редактирование, расчет, продажа). Число блоков на одном уровне иерархии – не более 6, иначе восприятие диаграмм будет затруднено. Число уровней иерархии не ограничено, но обычно – не более 5. Блоки в диаграммах IDEF0 связаны

дугами (стрелками), которые отображают множества объектов (данных). Управление (*control*) определяет условия выполнения. Примеры управления: требования, чертеж, стандарт, указания, план. Механизм (*mechanism*) выражает используемые средства, например: компьютер, САПР, оснастка, заказчик, фирма. Входы и выходы могут быть любыми объектами.

Пример диаграммы IDEF0 показан на рис. 6.2, где представлены функции, выполняемые на начальных этапах процесса реинжиниринга предприятия. На диаграмме показаны четыре этапа подготовки к реинжинирингу.

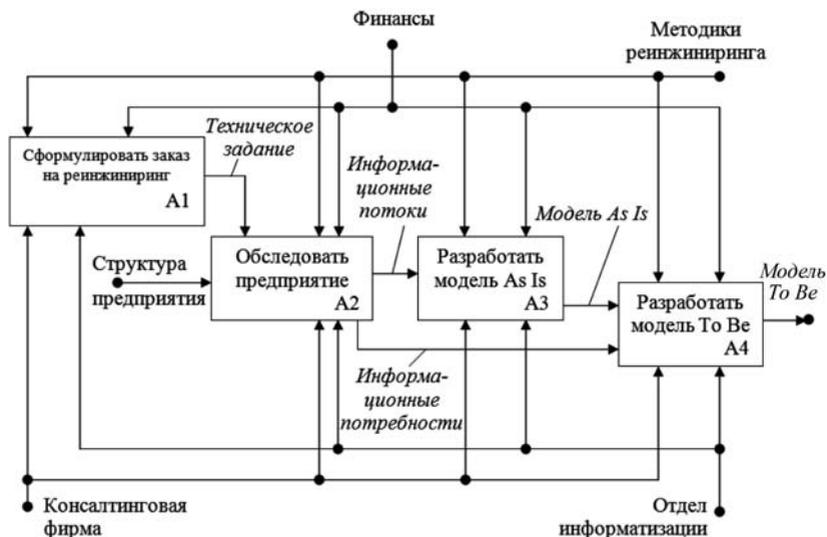


Рис. 6.2. Пример IDEF0-диаграммы

После формулировки заказа на перестройку процессов функционирования на базе информатизации управления производится обследование предприятия, выявляются его структура, информационные потоки между подразделениями, внешние информационные связи, степень компьютеризации, наличие вычислительной сети и т.п. На основании полученных данных составляется функциональная модель As Is и разрабатывается

модель To Be. Эта модель верхнего уровня далее конкретизируется, каждая из функций раскрывается более подробно на диаграммах следующих уровней.

Методика информационного проектирования приложений IDEF1X основана на построении информационных моделей приложений в виде диаграмм «сущность – связь». Для описания сущностей и отношений используется язык диаграмм.

Сущности в IDEF1X-диаграммах изображаются в виде прямоугольников, отношения – в виде стрелок. Отношения между сущностями в IDEF1X являются бинарными. Выделяют *идентифицирующие отношения* – связи типа «родитель – потомок», в которых потомок (зависимая сущность) однозначно определяется своей связью с родителем, и *неидентифицирующие отношения*, означающие, что у связанного этим отношением экземпляра одной сущности может быть (а может и не быть) соответствующий экземпляр второй сущности. Идентифицирующее отношение изображают на диаграмме сплошной линией между прямоугольниками связанных сущностей, неидентифицирующее – пунктирной линией. На дочернем конце линии должно быть утолщение (жирная точка). Мощность k связи – число экземпляров зависимой сущности, соответствующее одному экземпляру родительской сущности. Известное значение мощности может быть указано около утолщенного конца линии связи. При этом символ p означает $k \geq 1$, а символу z соответствует $k = 0$ или 1. Отсутствие символа интерпретируется как $k \geq 0$.

Различают также специфические и неспецифические отношения. *Неспецифические отношения* – это связи типа «многие ко многим», они обозначаются сплошной линией с утолщениями на обоих концах.

В отношениях «родитель – потомок» возможно наличие у потомка единственного родителя или нескольких родителей (ассоциативная связь). Выделяют также отношения категоризации (наследования), отражающие связи между некоторой общей сущностью и вариантами ее реализации (категориями). Примером категориальной связи является отношение «тип прибора – альтернативные варианты этого прибора».

Среди атрибутов различают ключевые и неключевые. Значение *ключевого атрибута* (ключа) однозначно идентифицирует экземпляр сущности. *Внешний ключ* – это атрибут (или атрибуты), входящий в ключ родителя и наследуемый потомком. На IDEF1X-диаграммах ключи записываются в верхней части прямоугольника сущности, причем внешние ключи помечают меткой FK (*Foreign Key*), неключевые атрибуты помещают в нижнюю часть прямоугольников. В идентифицирующих отношениях все ключи родителя входят и в ключи потомка, в неидентифицирующих – ключи родителя относятся к неключевым атрибутам потомка.

Между IDEF0 и IDEF1X-моделями одного и того же приложения существуют определенные связи. Так, стрелкам на IDEF0-диаграммах соответствуют атрибуты некоторых сущностей в IDEF1X-моделях, что нужно учитывать при построении информационных моделей.

Поведенческие аспекты приложений отражает методика IDEF3. Если методика IDEF0 связана с функциональными аспектами и позволяет отвечать на вопросы «Что делает система?», то в IDEF3 детализируются и конкретизируются IDEF0-функции и IDEF3-модель отвечает на вопросы «Как система это делает?». Язык IDEF3 – язык диаграмм, помогающий разработчику моделей наглядно представить моделируемые процессы. В IDEF3 входят два типа описаний: 1) процессно-ориентированные в виде последовательности операций; 2) объектно-ориентированные, выражаемые диаграммами перехода, характерными для конечно-автоматных моделей [3]. Программный продукт ProSim реализует методику IDEF3.

В заключение этого раздела рассмотрим все семейство стандартов IDEF. Применяемые в CASE-средствах разные методики и модели описывают различные свойства систем, важные, например, с точки зрения их автоматизации, а также позволяющие количественно оценить параметры проектов. Следует отметить, что спектр свойств систем различного назначения очень широк, и не все они к настоящему времени отражены в адекватных моделях. В то же время для класса информационных систем

организационного типа (*Management Information Systems – MIS*) адекватные модели разработаны и поддерживаются соответствующими средствами автоматизации.

Взаимная совокупность методик и моделей концептуального проектирования IDEF (*Integrated DEFINition*) разработана в США по программе *Integrated Computer Aided Manufacturing*. В настоящее время имеются методики функционального, информационного и поведенческого моделирования и проектирования, в которые входят IDEF-модели, приведенные ниже.

Название	Назначение
IDEF0	Функциональное моделирование <i>Function Modeling Method</i>
IDEF1 и IDEF1X	Информационное моделирование <i>Information and Data Modeling Method</i>
IDEF2	Поведенческое моделирование <i>Simulation Modeling Method</i>
IDEF3	Моделирование деятельности <i>Process Flow and Object State Description Capture Method</i>
IDEF4	Объектно-ориентированное проектирование <i>Object-oriented Design Method</i>
IDEF5	Систематизация объектов приложения <i>Ontology Description Capture Method</i>
IDEF6	Использование рационального опыта проектирования <i>Design Rational Capture Method</i>
IDEF8	Взаимодействие человека и системы <i>Human-System Interaction Design</i>
IDEF9	Учет условий и ограничений <i>Business Constraint Discovery</i>
IDEF14	Моделирование вычислительных сетей <i>Network Design</i>

IDEF0 реализует методику функционального моделирования сложных систем. Наиболее известной реализацией IDEF0 является методология SADT (*Structured Analysis and Design Technique*), предложенная в 1973 г. Д. Россом и впоследствии ставшая осно-

вой стандарта IDEF0. Эта методика рекомендуется для начальных стадий проектирования.

IDEF1X и **IDEF1** реализуют методики инфологического проектирования баз данных. В IDEF1X имеется ясный графический язык для описания объектов и отношений в приложениях, так называемый язык диаграмм «сущность – связь» (ERD – *Entity-Relations Diagrams*). Разработка информационной модели по IDEF1X выполняется в несколько этапов:

- выясняются цели проекта, составляется план сбора информации, при этом обычно исходные положения для информационной модели следуют из IDEF0-модели;
- выявляются и определяются основные сущности – элементы базы данных, в которых будут храниться эти системы;
- выявляются и определяются основные отношения, результаты представляются графически в виде так называемых ER-диаграмм;
- детализируются нестандартные отношения, определяются ключевые атрибуты сущностей; детализация отношений заключается в замене связей «многие ко многим» на связи «многие к одному» и «один ко многим»;
- определяются атрибуты сущностей.

IDEF2 и **IDEF3** реализуют поведенческое моделирование. Если методика IDEF0 связана с функциональными аспектами и позволяет отвечать на вопрос «Что делает система?», то в этих методиках детализируется ответ «Как система это делает». В основе поведенческого моделирования лежат модели и методы имитационного моделирования систем массового обслуживания, сети Петри, возможно применение модели конечного автомата, описывающей поведение системы как последовательности смен состояний.

Перечисленные методики относятся к так называемым структурным методам.

IDEF4 реализует объектно-ориентированный анализ больших систем. Он предоставляет пользователю графический язык для изображения классов, диаграмм наследования, таксономии методов.

IDEF5 направлен на представление онтологической информации приложения в удобном для пользователя виде. Для этого используются символические обозначения (дескрипторы) объектов, их ассоциаций, ситуаций и схемный язык описания отношений классификации, «часть-целое», перехода и т. п. В методике имеются правила связывания объектов (термов) в предложения и аксиомы интерпретации термов.

IDEF6 направлен на сохранение рационального опыта проектирования информационных систем, что способствует предотвращению структурных ошибок.

IDEF8 предназначен для проектирования диалогов человека и технической системы.

IDEF9 используется для анализа имеющихся условий и ограничений (в том числе физических, юридических, политических) и их влияния на принимаемые решения в процессе реинжиниринга.

IDEF14 предназначен для представления и анализа данных при проектировании вычислительных сетей на графическом языке с описанием конфигураций, очередей, сетевых компонентов, требований к надежности и т. п.

Контрольные вопросы

1. Дайте определение понятия «концептуальное проектирование» (модели «As Is» и «To Be»).

2. Опишите функциональную и информационную модели производственного процесса (модели «To Be»).

3. Опишите поведенческую и структурную модели производственной системы.

4. Перечислите общие положения, на которых концептуально базируется современный подход к созданию и информационному обеспечению сложных процессов и систем.

5. Раскройте понятие «совмещенное (параллельное)» проектирование. Приведите пример.

6. В чем проявляется синергетический характер мехатронных систем?

7. Объясните причины широкого применения методов визуализации и использования «встроенных» систем при проектировании сложных систем.

8. Чем объясняется необходимость проектирования интеллектуальных систем управления сложными системами?

9. Раскройте двойное толкование аббревиатуры CASE.

10. Рассмотрите кратко методологию SADT.

11. Укажите основное назначение методик IDEF0, IDEF1X, IDEF3.

12. Опишите процесс построения функциональных моделей в процессе проектирования сложных систем (с использованием IDEF0-методики).

13. В чем состоит суть методики информационного проектирования сложных систем (приложений) IDEF1X?

14. Какие аспекты проектирования отражает методика IDEF3?

15. Охарактеризуйте все семейство стандартов IDEF.

16. Перечислите методики IDEF, относящиеся к структурным методам проектирования.

Лекция 7

ИНФОРМАЦИОННАЯ ПОДДЕРЖКА ЖИЗНЕННОГО ЦИКЛА МЕХАТРОННЫХ СИСТЕМ

Описанные выше методы IDEF задают единообразный подход к моделированию приложений, но не затрагивают проблем единообразного представления данных в процессах информационного обмена между разными компьютерными системами и приложениями. Необходимость решения этих проблем в интегрированных CAD/CAM/CAE-системах привела к появлению ряда унифицированных форматов представления данных в межкомпьютерных обменах, среди которых наиболее известными в машиностроении являются форматы IGES, DXF (см. раздел 5.2), EDIF в электронике и другие. Ограниченные возможности этих стандартов обусловили развитие работ в направлении полной информации обо всем жизненном цикле продукта, включая проектирование, анализ производства, контроль качества, испытание и обслуживание, помимо обычных данных технических требований, и привели к появлению формата STEP (Standard for Exchange of Product data – стандарт обмена данными о модели продукта) в составе CALS-технологий.

7.1. CALS-технологии (основные понятия) [1, 3]

В настоящее время термин CALS (Continuous Acquisition and Life Cycle Support) как «Непрерывное сопровождение и информационная поддержка всех этапов жизненного цикла изделий (ЖЦ)». На русском языке понятию CALS соответствует ИПИ (информационная поддержка изделий) или КСПИ (компьютерное сопровождение и поддержка изделий).

Из этих определений следует, что CALS-технологиями называются технологии комплексной компьютеризации всех сфер промышленного производства, цель которых – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла (см. рис. 1.1). Основные спецификации представлены проектной, технологической, производственной, маркетинговой и эксплуатационной документацией. В CALS-системах предусмотрены хранение, обработка и передача информации в компьютерных средах, оперативный доступ к данным в нужное время и в нужном месте.

Стратегия CALS предполагает создание единого информационного пространства (ЕИП) для всех участников ЖЦ изделия. ЕИП предполагает представление всей информации об изделии в электронном виде и организацию совместного использования информации для всех участников ЖЦ изделия (в соответствии с правилами доступа).

Основные свойства ЕИП:

- информация предоставляется в электронном виде;
- охватывается вся информация об изделии, созданная всеми исполнителями на любом этапе ЖЦ;
- ЕИП выступает основным источником данных для исполнителей, предоставляя (в соответствии с правами доступа) нужную информацию в нужное время и в нужном виде;
- для интеграции информации в ЕИП используются только международные и отраслевые стандарты, поддерживаемые подавляющим большинством производителей прикладных систем;
- для создания ЕИП используются существующие на предприятиях программно-аппаратные средства; необходима адаптация этих систем к работе в рамках ЕИП.

Стратегия CALS предусматривает два этапа при переходе к ЕИП:

- автоматизация отдельных процессов (или этапов) ЖЦ изделия и представление данных на них в электронном виде;
- интеграция автоматизированных процессов и относящихся к ним данных, уже представленных в электронном виде, в рамках ЕИП.

CALS-технологии – это набор методов реализации стратегии CALS:

- технологии реинжиниринга бизнес-процессов – набор методов реструктуризации бизнес-процессов в целях повышения их эффективности;

- технологии представления данных об изделии в электронном виде – набор методов для представления в электронном виде данных об изделии, относящихся к отдельным процессам ЖЦ изделия. Эти данные являются информационными моделями, обеспечивающими автоматизацию отдельных процессов ЖЦ изделия. Технологии представления данных включают в себя также технологии перевода данных из бумажного в электронный вид (*E-commerce* – *E-бизнес*). Рассматриваемая группа CALS-технологий состоит из известных методов, реализованных в соответствующих автоматизированных системах;

- технологии интеграции данных об изделии в рамках ЕИП – набор методов для интеграции автоматизированных процессов ЖЦ и относящихся к ним данных, представленных в электронном виде, в рамках ЕИП; эти технологии реализуются с помощью класса автоматизированных систем, называемых системами управления данными об изделии (проектными данными) – PDM (*Product data Management*). Для реализации технологии PDM существуют специализированные программные средства, называемые PDM-системами. Пользователями PDM-систем являются конструкторы, технологи, а также сотрудники, работающие в других сферах производства и обслуживания – продажи, маркетинга, снабжения, финансов, сервиса, эксплуатации и т. п.

Главная задача создания и внедрения CALS-технологий – обеспечение единообразных описаний и интерпретации данных независимо от места и времени их получения в общей системе, имеющей масштабы вплоть до глобальных. Структура проектной, технологической и эксплуатационной документации, языки ее представления должны быть стандартизованными. Тогда становится реальной успешная работа над общим проектом разных коллективов, разделенных во времени и пространстве и применяющих разные системы CAE/CAD/CAM. Одна и та же

конструкторская документация может быть использована многократно в разных проектах, а одна и та же технологическая документация – адаптирована к разным производственным условиям, что позволяет существенно сократить и удешевить общий цикл проектирования и производства. Кроме того, упрощается эксплуатация систем.

CALS-технологии не отвергают существующие автоматизированные системы проектирования и управления, а являются средством их эффективного взаимодействия. Поэтому интеграция автоматизированных систем на современных предприятиях должна быть основана на CALS-технологиях. Внедрение их требует освоения имеющихся технологий и CALS-стандартов, развития моделей, методов и программ автоматизированного проектирования и управления. Важные проблемы, требующие решения при создании CALS-систем, – управление сложностью проектов и интеграция программного обеспечения, включая вопросы декомпозиции проектов, распараллеливания проектных работ, целостности данных, межпрограммных интерфейсов и др.

CALS-технологии зародились в 1980-е гг. в недрах военно-промышленного комплекса США в связи с их планами в области стратегической оборонной инициативы (СОИ). Поэтому не удивительно, что среди имеющихся CALS-стандартов фигурирует большое число стандартов и рекомендаций DoD (Министерства обороны США). Для реализации планов СОИ требовались совместные усилия многих промышленных компаний и предприятий в проектировании, производстве и логистической поддержке сложных изделий, а это означало необходимость унификации представления данных об изделиях. Было осознано, что для взаимодействия автоматизированных систем разных предприятий нужна унификация не только формы, но и содержания (семантики) проектной, технологической, эксплуатационной и другой информации о совместно производимой продукции. Другими словами, требовалось создание единой информационной среды взаимодействия всех крупнейших фирм американского военно-промышленного комплекса.

Оказалось, что это чрезвычайно сложная проблема, решение которой требует длительной и многосторонней проработки в масштабах, выходящих за пределы одной страны. Выяснилось также, что создание единой информационной среды требуется не только для уникальных программ типа СОИ, но и для производства любых сложных систем, в первую очередь военной техники, если ее производство основано на взаимодействии многих предприятий.

В связи с возникшими практическими потребностями рядом комиссий и комитетов в рамках международных организаций были начаты работы по созданию информационных технологий и взаимодействию предприятий и выражающих их международных стандартов. Например, в Международной организации стандартизации (*International Standard Organization – ISO*) этими вопросами ведает подкомитет SC4 комитета TC184. В SC4 имеется несколько рабочих групп, занимающихся конкретными сериями стандартов. В настоящее время в ведущих индустриальных странах мира созданы национальные органы, координирующие работу в области CALS-технологий. В международном масштабе развитием CALS помимо ISO занимаются и такие организации, как ICC (*International CALS-Congress*), EIA (*Electronics Industry Association*), IEEE (*Institute of Electrical and Electronics Engineers*) и др. В России в рамках Госстандарта создан технический комитет № 431 «CALS-технологии».

В 1990-х гг. разработан и к настоящему времени принят ряд серий международных стандартов, представляющих CALS-технологии, среди которых наиболее значимы стандарты ISO 10303 STEP (*Standard for Exchange of Product Data*). В контрактах, заключаемых на поставку зарубежным заказчикам военной техники, требования к изделиям и документации на них, как правило, формулируются с позиций международных CALS-стандартов и стандартов DoD (последние имеют префикс MIL).

Развитие CALS-технологий стимулирует образование виртуальных производств, при которых процесс создания спецификаций с информацией для программного управляемого технологического оборудования, достаточной для изготовления изделия,

может быть распределен во времени и пространстве между многими организационно автономными проектными организациями.

Ожидается, что успех на рынках сложной технической продукции будет немислим вне CALS-технологий. Так, уже сегодня фирмы, предлагающие военную технику без электронной документации, выполненной в соответствии с CALS-стандартами, не имеют никаких шансов на успех в конкурентной борьбе.

Итак, CALS-технологии призваны, в конечном счете, повысить эффективность создания и использования сложной техники (в том числе мехатронных систем). В чем выражается повышение эффективности?

Во-первых, улучшается качество изделий за счет более полного учета имеющейся информации при проектировании и принятии управленческих решений. Так, обоснованность решений, принимаемых в автоматизированной системе управления предприятием (АСУП), будет выше, если лицо, принимающее решение, и соответствующие программы АСУП имеют оперативный доступ не только к базе данных АСУП, но и к базам данных других автоматизированных систем – системам автоматизированного проектирования (САПР), автоматизированной системе технологической подготовки производства (АСТПП) и автоматизированной системе управления технологическими процессами (АСУТП), и, следовательно, могут оптимизировать планы работ, содержание заявок, распределение исполнителей, выделение финансов и т.п. При этом под оперативным доступом необходимо понимать не просто возможность считывания данных из баз данных, но и легкость их правильной интерпретации, т.е. согласованность по синтаксису и семантике с протоколами, принятыми в АСУП. То же относится и к другим системам, например, технологические подсистемы должны с необходимостью воспринимать и правильно интерпретировать данные, поступающие от подсистем автоматизированного конструирования. Последнего не так легко добиться, если основное предприятие и организации-смежники работают с разными автоматизированными системами.

Во-вторых, сокращаются материальные и временные затраты на проектирование и изготовление изделий. Применение

CALS-технологий позволяет существенно сократить объемы проектных работ, так как описания ранее выполненных удачных разработок компонентов и устройств, многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю CALS-технологий. Доступность опять же обеспечивается согласованностью форматов, способов, руководств в разных частях общей интегрированной системы. Кроме того, появляются более широкие возможности для специализации предприятий, вплоть до создания виртуальных предприятий, что также способствует снижению затрат.

В-третьих, значительно снижаются затраты на эксплуатацию благодаря реализации функций интегрированной логистической поддержки. Существенно облегчается решение проблем ремонтно-пригодности, интеграции продукции в различного рода системы и среды, адаптации к меняющимся условиям эксплуатации и т. п.

По аналогии с САПР для CALS-технологий разработаны различные виды обеспечения их применения: *лингвистическое, информационное, программное, математическое, методическое, техническое и организационное.*

К **лингвистическому обеспечению** относятся языки и форматы данных о промышленных изделиях и процессах, используемые для представления и обмена информацией на этапах жизненного цикла изделий.

Информационное обеспечение составляют базы данных, в которых имеются сведения о промышленных изделиях, используемые разными системами в процессе проектирования, производства, эксплуатации и утилизации продукции. В состав информационного обеспечения входят также серии международных и национальных CALS-стандартов и спецификаций.

Программное обеспечение CALS представлено программными комплексами, предназначенными для поддержки единого информационного пространства этапов жизненного цикла изделий. Это, прежде всего, системы управления документами и документооборотом, управления проектными данными (PDM), взаимодействия предприятий в совместном электронном бизнесе

(СРС), подготовки интерактивных электронных технических руководств и некоторые другие.

Математическое обеспечение CALS включает методы и алгоритмы создания и использования моделей взаимодействия различных систем в CALS-технологиях. Среди этих методов в первую очередь следует назвать методы имитационного моделирования сложных систем, методы планирования процессов и распределения ресурсов.

Методическое обеспечение CALS представлено методиками выполнения таких процессов, как параллельное (совмещенное) проектирование и производство, структурирование сложных объектов, их функциональное и информационное моделирование, объектно-ориентированное проектирование, создание антологий приложений.

К **техническому обеспечению** CALS относят аппаратные средства получения, хранения, обработки и визуализации данных при информационном сопровождении изделий. Взаимодействие частей виртуальных предприятий, систем, поддерживающих разные этапы жизненного цикла изделий, происходит через линии передачи данных и сетевое коммутирующее оборудование. Однако используемые технические средства не являются специфическими для CALS, и поэтому вопросы технического обеспечения не рассматриваются.

Наконец, организационное обеспечение **организационное обеспечение** CALS представлено различного рода документами, совокупностью соглашений и инструкций, регламентирующих роли и обязанности участников жизненного цикла промышленных изделий.

За последний год появился целый ряд предприятий, которые активно занимаются изучением и применением CALS-технологий для решения конкретных производственных задач повышения качества и конкурентоспособности выпускаемой продукции. К таким организациям относятся АВПК «Сухой», АНТК им. Туполева, ОАО «Туламашзавод», Конструкторское бюро приборостроения (г. Тула), корпорация «Компомаш», корпорация «Метран» и др. Разработкой конкретных применений CALS-технологий для

ряда областей промышленности занимается МГТУ «Станкин». Решением координационного совета РАН по техническим наукам от 12 февраля 2002 г. Институту конструкторско-технологической информатики РАН поручена координация работ по CALS (ИПИ)-технологиям.

7.2. STEP-стандарты [5]

Выше было отмечено, что для реализации CALS-технологий необходимо обеспечить единообразное описание и интерпретацию данных, независимо от места и времени их получения в общей системе, имеющей масштабы вплоть до глобальных. Это единообразие достигается за счет разработки системы CALS-стандартов, созданных под эгидой ISO (*International Standard Organization*) – международной организации стандартизации.

Центральное место в системе CALS-стандартов занимает ISO 10303 под названием STEP (*Standard for Exchange of Product Data* – стандарт для обмена данными о промышленных изделиях), определяющий средства описания (моделирования) промышленных изделий на всех стадиях их ЖЦ. Для этого был разработан инвариантный к приложениям язык Express, введенный в STEP. Совокупность стандартов STEP составляет основу CALS-технологий.

В STEP используются следующие важные понятия:

AAM – *Application Activity Model* – функциональная модель IDEFO для определенного приложения;

ARM – *Application Requirements Model* – модель, представляющая данные с точки зрения пользователя. В частности, в этой модели данные могут быть выражены как средствами, типичными для приложения, так и с использованием синтаксиса языка Express;

AIM – *Application Interpreted Model* – ARM-модель, переведенная в STEP-представление с использованием ряда унифицированных в STEP-понятий, закрепленных в интегрированных ресурсах;

AP – *Application Protocol* – STEP-стандарт, отражающий специфику конкретного приложения;

SDAI – *Standard Data Access Interface* – программный интерфейс к источникам данных (репозиториям) прикладных систем (в том числе к библиотекам моделей систем CAD/CAM) с переводом моделей в STEP- файлы; используется в STEP-средах для организации обменов между приложениями через общую базу данных STEP.

STEP – это совокупность стандартов, состоящих из нескольких томов. Тома имеют свои номера и обозначаются как «часть №» или ISO 10303-№. К настоящему времени разработано более сотни томов. Стандарты ISO 10303 определяют средства описания (моделирования) промышленных изделий на всех этапах их ЖЦ.

Том 1 (ISO 10303-1) – вводный стандарт, выполняющий роль аннотаций всей совокупности томов. В этом стандарте вводится ряд терминов, используемых в других стандартах, например таких, как продукт (*product*), приложение (*application*), проектные данные (*product data*), модель (*model*), модели AAM, AIM, ARM, прикладной протокол (AP), интегрированный ресурс (*integrated resource*), элемент функциональности (*unit of functionality* – UoF).

Томы 11–14 – методы описания (*description methods*).

Томы 21–29 – методы реализации (*implementation methods*).

Томы 31–35 – основы тестирования моделей (*conformance testing methodology and framework*).

Томы 41–50 – интегрированные основные ресурсы (*integrated generic resources*).

Томы 101–108 – интегрированные прикладные ресурсы (*integrated application resources*).

Томы 201–236 – прикладные протоколы (*application protocols*).

Томы 301–332 – абстрактные тестовые наборы (*abstract test suites*).

Томы 501–520 – прикладные компоненты (*application interpreted constructs*).

Ряд томов переведен на русский язык и представлен в виде национальных стандартов России. Это, например, ГОСТ Р ИСО 10303-1-99, посвященный обзору и основополагающим принципам STEP, ГОСТ Р ИСО 10303-11-99 – справочное руководство по языку Express, ГОСТ Р ИСО 10303-21-99 – то же по обменному файлу, ГОСТ Р ИСО 10303-41-99 – описание интегрированных

родовых ресурсов. Перечисленные документы соответствуют стандартам ISO 10303-1, ISO 10303-11, ISO 10303-21, ISO 10303-41. Подготовлены к утверждению ГОСТы, соответствующие томам 43, 44, 203 стандарта ISO 10303.

Таким образом, в томах STEP описаны основные принципы обмена данными, изложены правила языка Express, даны методы его реализации. Модели, методы тестирования моделей, ресурсы как общие для приложений, так и некоторые специальные (например, геометрические и топологические модели, описание материалов, процедуры черчения, конечно-элементного анализа и т. п.), введены прикладные протоколы, отражающие специфику моделей в конкретных предметных областях. Вопросы взаимодействия АС в STEP уделяется основное внимание – выделена подгруппа томов, посвященных способам обмена данными между разными системами, созданными в рамках STEP-технологий.

Развитие CALS-технологий находит выражение также в разработке серий стандартов ISO13584 *Parts Library* (сокращенно *P_Lib*), ISO 14959 *Parametrics*, ISO15531 *Manufacturing Management Data (Mandate)*, ISO 18876 *Integration of Industrial Data for Exchange, Access, and Sharing (IIDEAS)*, ISO 8879 *Standard Generalized Markup Language (SGML)* (см. т. 107, 108).

Более подробно со структурой STEP-стандартов (методами описания диалектов языка Express № 11–19, методами реализации межпрограммного информационного обмена между прикладными системами в STEP-среде № 21–29, прикладными протоколами № 201–236, интегрированными ресурсами и прикладными компонентами № 41–50 и № 101–108, № 501–520) можно познакомиться в работе [5, с. 166–180].

7.3. Организация в STEP информационных обменов [1]

Возможны обмены через обменный файл, описанный выше (см. п. 5.2), и через базу данных SDAI (Standard Data Access Interface) – интерфейс к данным, представленным в соответствии с CALS-стандартами.

Обменный файл используется при связи моделей *A* и *B*, имеющих общие данные с различными обозначениями. Пользователь должен написать перекодировщик (например, на языке Express-X), с помощью которого отождествляются идентификаторы одних и тех же сущностей, имевших разные обозначения в моделях *A* и *B* (рис. 7.1).

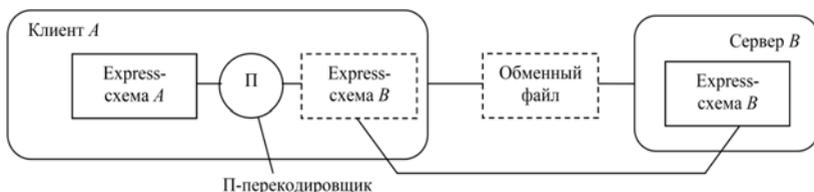


Рис. 7.1. Взаимодействие Express-приложений через обменный файл

Связь через интерфейс SDAI отличается от предыдущего способа обмена тем, что в SDAI имеет место не просто обмен, а разделение данных многими пользователями, и SDAI фактически выступает в роли метамодели для разных САПР. Другими словами, SDAI представляет собой интерфейс, содержащий набор функций на языках C++ и C, для доступа к разделяемым моделям, которые могут быть представлены в виде обменного файла (рис. 7.2).

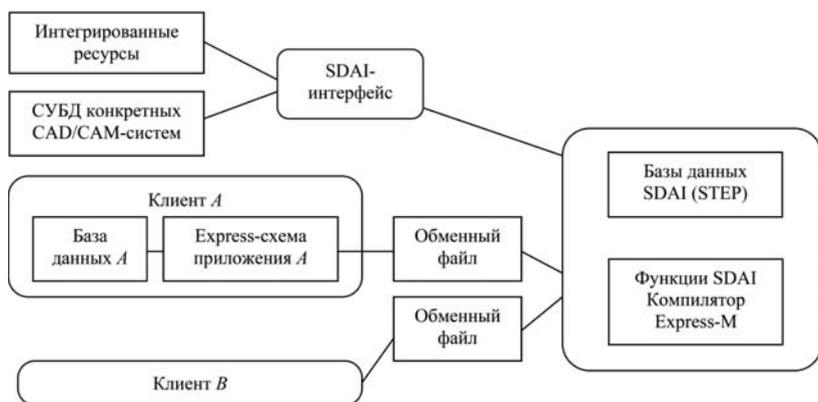


Рис. 7.2. Взаимодействие Express-приложений через базу данных SDAI

В CALS-технологиях рассматриваются не только вопросы представления данных и организации информационных обменов, но и вопросы моделирования приложений и проектирования сложных систем на базе концептуального проектирования.

В заключение данной лекции рассмотрим проблемы практического использования CALS-технологий.

7.4. Проблемы практического использования CALS-технологий

Проблема внедрения CALS-технологий в практику работы промышленных предприятий имеет организационный, технологический, кадровый аспекты.

В организационном плане прежде всего необходима убежденность руководителей предприятий в целесообразности (а в ряде случаев и в неизбежности) перехода к CALS-технологиям, что позволит сконцентрировать усилия на разработке и реализации планов реинжиниринга предприятий с постепенным внедрением элементов CALS-технологий. Эти планы направлены на создание корпоративной автоматизированной системы (КАС), интегрирующей системы автоматизации проектирования, управления и технологической подготовки производства. Особую важность вопросы создания КАС на базе CALS-технологий имеют для производственных объединений, включающих несколько предприятий.

Автономно функционирующие системы САПР, АСУП, АСУТП уже имеются на многих предприятиях. Очевидно, что при их интеграции на основе CALS-технологий собственно задача замены уже используемых систем какими-либо другими не ставится, поскольку замена – процедура дорогостоящая, длительная и болезненная, требующая переучивания специалистов. Замена может быть вызвана не особенностями CALS, а лишь выявленной недостаточной функциональностью используемых систем (например, невозможностью проектирования сборок с тысячами компонентов).

Что в этом случае подразумевается под «единой электронной моделью изделия»? Ответ заключается в следующем: единая электронная модель изделия имеет место, если любая подсистема или пользователь КАС (с соответствующими полномочиями) может, обратившись к базе данных системы, получить нужное подмножество данных об изделии, причем это подмножество будет согласовано с подмножествами, запрашиваемыми другими подсистемами или пользователями. Очевидно, что подмножества данных (профили), запрашиваемые конструкторскими, технологическими, управленческими и другими подсистемами и службами, включают как специфические, так и общие атрибуты, причем общие атрибуты будут иметь одинаковые значения в каждом подмножестве.

Если на предприятии автоматизация была развита слабо, то создание на нем КАС нужно начинать с обследования деятельности предприятия. Перед обследованием формируются и в процессе его проведения уточняются цели обследования – определение возможностей и ресурсов для повышения эффективности функционирования предприятия на основе автоматизации процессов управления, проектирования, документооборота и т. п. Содержание обследования – выявление структуры предприятия, выполняемых функций, информационных потоков, имеющихся опыта и средств автоматизации. Обследование проводят системные аналитики (системные интеграторы) совместно с представителями организации-заказчика.

На основе анализа результатов обследования строят модель, отражающую деятельность предприятия на данный момент. Такую модель называют As Is – «как есть». Далее разрабатывают исходную концепцию КАС. Эта концепция включает в себя предложения по изменению структуры предприятия, взаимодействию подразделений, информационным потокам, что выражается в модели To Be – «как должно быть».

Результаты анализа конкретизируются в техническом задании на создание КАС. В нем указывают потоки входной информации, типы выходных документов и предоставляемых услуг, уровень защиты информации, требования к производительности

(пропускной способности) и т. п. Техническое задание направляют заказчику для обсуждения и окончательного согласования.

Перевод документов в электронную форму – одна из первых задач внедрения CALS-технологий. Для создаваемого электронного архива документов нужно в соответствии со спецификациями MIL-PRF-2800х выбрать единые форматы представления чертежной и текстовой информации. Далее необходимы разработка проекта корпоративной вычислительной сети и его реализация, выбор, приобретение и установка системы PDM, обучение сотрудников предприятия работе в КАС.

Для успешного внедрения и использования CALS-технологий предприятие должно располагать инженерными кадрами, знающими методы и умеющими использовать средства автоматизированного проектирования, поскольку CALS-технологии развиваются прежде всего на базе САПР. В настоящее время в технических вузах страны студенты изучают такие разделы САПР, как геометрическое моделирование и машинная графика, математическое моделирование, оптимизация проектных решений. Целесообразно дополнить учебные программы соответствующих курсов, включив в них изучение вопросов CALS-технологий.

Так, вопросы геометрического моделирования, изучаемые в курсах инженерной графики, основ САПР или в специальном курсе «Вычислительная геометрия», полезно дополнить фрагментами соответствующих прикладных протоколов STEP. В разделах, посвященных математическому моделированию, нужно знакомить студентов с возможностями многоаспектного моделирования, в том числе на базе языка VHDL-AMS. Студенты должны быть знакомы с методиками концептуального проектирования IDEF0, IDEF1X и объектно-ориентированного проектирования на базе языка UML. Эти вопросы, так же как структура стандартов STEP и основы языка Express, могут быть предметом изучения в отдельном курсе «CALS-технологии» или в курсе «Основы автоматизированного проектирования». В раздел оптимизации проектных решений нужно включить методы оптимального планирования и распределения ресурсов. Наконец, следует предусмотреть знакомство студентов с основами Internet-технологий,

в том числе с представлением документов с помощью языков разметки.

Естественно, что преподавание вопросов CALS-технологий требует развития лабораторной базы. В составе программного обеспечения, поддерживающего лабораторные циклы по САПР и CALS, необходимо иметь по крайней мере один из пакетов MCAD (с модулями CAD и CAM), программы моделирования (например, типа ПА9), пакеты поддержки методик функционального и информационного моделирования (например, BPWin и ERWin), XML-редактор, желательно приобретение PDM. Циклы дисциплин радиоэлектронного профиля должны поддерживаться программным обеспечением сквозного проектирования радиоэлектронной аппаратуры и в первую очередь программами разработки схем на базе ПЛИС и проектирования печатных плат.

Более глубоко с отдельными разделами CALS-технологий: обзор CALS-стандартов, математическое, лингвистическое и программное обеспечение – можно ознакомиться в [1, 3, 7, 12].

Контрольные вопросы

1. Дайте обоснование необходимости появления формата STEP обмена данными.
2. Дайте общее определение термина CALS-технологии.
3. Сформулируйте суть стратегии CALS.
4. Перечислите основные свойства единого информационного пространства (ЕИП).
5. Опишите два этапа перехода к ЕИП.
6. Перечислите методы реализации стратегии CALS.
7. Сформулируйте главную задачу создания и внедрения CALS-технологий.
8. Опишите историю зарождения CALS-технологий.
9. Укажите причины повышения эффективности создания и применения сложной техники при использовании CALS-технологий.
10. Опишите различные виды обеспечения CALS-технологий.

11. Укажите условия, необходимые для реализации CALS-технологий.

12. Сформулируйте основное назначение стандарта ISO 10303 (STEP-стандарт).

13. Дайте общую характеристику совокупности стандартов ISO 10303 (содержание томов, входящих в стандарты).

14. Опишите два способа организации в STEP информационных обменов.

15. Опишите порядок внедрения CALS-технологий в практику работы промышленных предприятий.

16. Опишите методику преподавания CALS-технологий для обучения инженерных кадров.

Лекция 8

ОСНОВНЫЕ ПРИКЛАДНЫЕ МЕТОДЫ И СРЕДСТВА ПОСТРОЕНИЯ ИНТЕГРИРОВАННЫХ МЕХАТРОННЫХ СИСТЕМ

8.1. Формирование общих проектных решений [3, 12, 13]

Исторически сложилось, что при традиционном проектировании управляемых машин и процессов разработка механической, электронной, информационной и компьютерной частей ведется последовательно и независимо друг от друга (рис. 8.1).

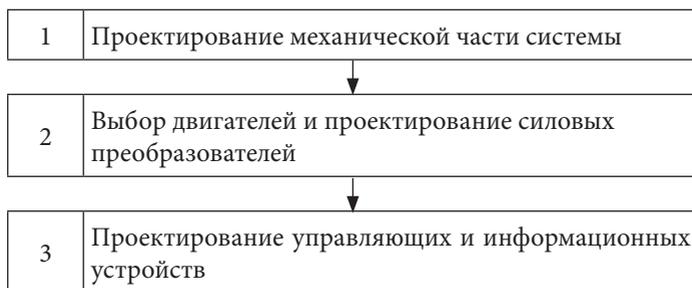


Рис. 8.1. Традиционный алгоритм проектирования

При этом возможности разработчика системы управления крайне ограничены, так как основные конструкторские решения уже приняты на предыдущих этапах, выбор двигателей и силовых преобразователей завершен и не подлежит корректировке при разработке электронной и управляющей частей системы. Кроме

того, при традиционном проектировании многочисленные интерфейсы, связывающие в мехатронной машине (производственном процессе) устройства различной физической природы (механические, электронные, информационные), предопределяют их конструктивную и аппаратно-программную сложность. Опыт эксплуатации сложных систем показывает, что до 70 % проблем их функционирования связаны с надежностью связей и соединений. Поэтому была предложена новая методологическая основа проектирования мехатронных систем – методы параллельного проектирования. Суть методологии параллельного проектирования заключается в одновременном и взаимосвязанном синтезе всех устройств мехатронной системы уже на этапах проектирования и изготовления интегрированных модулей системы.

Следуя [13], можно выделить следующие уровни интеграции в мехатронных системах: первый уровень образуют мехатронные (технологические) устройства и составляющие их элементы, второй – включает интегрированные мехатронные модули, третий – предполагает компоновку из мехатронных модулей, четвертый – предполагает построение на единой интеграционной платформе производственных систем (реконфигурируемых производств).

Описанные выше методы параллельного проектирования и выделения уровней интеграции в мехатронных системах позволяют определить направление и глубину интеграционных процессов при проектировании сложных систем.

Использование гибких технологий при проектировании и изготовлении мехатронных систем определяет два направления проектных и конструкторских работ:

- 1) собственно проектирование производственных процессов на предприятиях на базе готовых программных и аппаратных (технологических) компонентов с помощью специальных инструментальных средств разработки приложений (например, CASE-средств);

- 2) проектирование упомянутых компонентов и инструментальных средств, ориентированных на многократное применение при разработке многих производственных процессов (систем).

Первое направление базируется на системотехнике, информационных технологиях и программных продуктах, инструментальных средствах разработки приложений (см. лекции 6, 7).

Второе направление в большей мере относится к области разработки математического и программного обеспечения (МО, ПО), методов анализа и синтеза проектных решений, технологий программирования на базе знаний и т. п.

В дальнейшем в лекции основное внимание будет уделено первому направлению работ – собственно проектированию машинных комплексов и производственных процессов на предприятиях.

Проектирование любой сложной системы начинается с формулирования целей, видения будущей системы и исходных технических требований. Затем согласно теории системного проектирования сложных систем выполняются:

- функционально-структурный анализ системы;
- синтез структуры системы;
- структурный анализ и оптимизация (по критерию сложности) отдельных технических подсистем;
- разработка проектно-конструкторской документации в целом на систему.

Функция системы имеет приоритетное значение по отношению к ее структуре, так как различные структуры могут с разной степенью эффективности реализовать одну и ту же функцию.

Заметим, что сложные технические системы различного назначения (интегрированные мехатронные модули и машины, производственные процессы, интегрированные автоматизированные системы и др.) имеют ряд особенностей, которые необходимо учитывать при их проектировании, производстве, реализации (технологической реализации производства) и эксплуатации.

На всех этапах жизненного цикла технологических изделий и производственных процессов (см. рис. 1.1) используются технологии реинжиниринга и параллельного проектирования, методика SADT и стандарты IDEF-технологий (IDEF0, IDEF3, DFD) для проектирования функциональных моделей системы через их поведение по отношению к внешним объектам и внешней среде.

Для структурного описания систем и связей между их структурными элементами используется методология IDEF1X (задающая структуру баз данных для обеспечения процессов), имитационное моделирование, виртуальная инженерия, системы принятия проектных решений, интегрированные системы CAD/CAM/CAE/PDM.

Из поддерживающих методологию IDEF-систем можно отметить BPWin, ERWin, OOWin, Design IDEF др., а также ряд систем автоматизированного проектирования и управления техническими системами и международных стандартов ISO (STEP-стандарты) (см. лекцию 1).

8.2. Построение модели процесса разработки мехатронных систем

Для описания процесса разработки мехатронных систем (МС) далее будем использовать принятую в САПР терминологию этапов разработки:

- 1) функциональное проектирование (определение функций МС на основе анализа исходных требований);
- 2) функционально-структурный анализ (выбор вариантов структурных решений системы по ее заданной функции);
- 3) конструкторское проектирование элементов и модулей МС по результатам структурно-конструктивного анализа;
- 4) разработка проектной и конструкторской документации на систему;
- 5) разработка производства МС (включая технологическую подготовку производства);
- 6) испытание МС;
- 7) доводка МС;
- 8) эксплуатация МС.

Основные этапы проектирования мехатронных систем

Первые три (из перечисленных выше) этапа разработки определяют цикл проектирования (см. рис. 1.1) мехатронных систем. На

этом цикле используются инструментальные средства и методы информационных технологий, которые определяют возможность и уровень разрабатываемых систем. Среди этих методов и средств CALS-технологии, CASE-средства и поддерживающие их инструментальные системы (BPWin, ERWin, Design IDEF и др.), технологии реинжиниринга и параллельного проектирования (методики SADT, IDEF), системные среды Meta САПР (Framework), универсальные CAD/CAM/CAE, PDM, SCADA и другие системы.

В настоящее время актуальной становится задача автоматизации проектирования не отдельных элементов и подсистем, а всего процесса проектирования, конструирования и производства мехатронных систем.

Основными чертами современной технологии создания САПР являются [17]:

- переход к автоматизации всего процесса проектирования в целом;
- обеспечение взаимодействия между автоматизированными системами, ведущими расчеты, разработку, изготовление и т. д.;
- обеспечение возможности адаптации системы к объекту проектирования и развития ее в процессе эксплуатации;
- широкое использование диалоговых режимов работы, процедур коллективного проектирования;
- развитие новой методологии проектирования, ориентированной на сквозное применение средств автоматизации в рамках интегрированных CAD/CAM/CAE-систем, на основе которых создаются «приложения» – специализированные САПР.

Структурная модель МС строится на базе функциональной модели МС, полученной с помощью IDEF-технологий (IDEF0, IDEF1X, DFD) и поддерживающих их систем (BPWin, ERWin, Design IDEF), а также САПР Спрут, имитационного моделирования, систем принятия проектных решений, CALS-технологий. С этой целью функциональную модель МС подвергают декомпозиции на отдельные узлы и детали для технических (конструкционных) мехатронных систем или на отдельные технологические операции для производственных процессов. Далее с помощью интерактивных процедур (например, использования диалоговых

средств), осуществляется оптимизация структурной модели и ее компонентов (см. лекцию 1). Возможность осуществления интерактивного метода конструирования является одним из самых важных требований этапа конструирования. Для автоматизации процесса конструирования должен быть (как минимум) разработан пакет конструирования, состоящий из CAD-систем, обеспечивающих выполнение всего объема конструкторских работ и приложений, интегрированных с выбранными CAD-системами.

Инструментальные средства проектирования мехатронных систем

Далее рассмотрим комплекс инструментальных средств, охватывающих процесс концептуального проектирования (первые три этапа разработки системы) мехатронных систем [16]. На рынке IDEF-систем фирмой Computer Associations предложены CASE-средства BPWin и ERWin. CASE-средство верхнего уровня BPWin поддерживает методологию IDEF0 (функциональная модель), IDEF3 (Workflow Diagram – сценарий бизнес-процесса, поведенческое моделирование) и DFD (Dataflow Diagrams – диаграммы потока данных, набор связанных друг с другом действий).

Функциональная модель предназначена для описания существующих процессов на предприятии (As Is-модели) и того, к чему надо стремиться (модель To Be). Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействие с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграмма декомпозиций). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. Каждая диаграмма проверяется экспертами предметной области, заказчиками и пользователями процесса. В случае необходимости (чтобы осветить специфические стороны технологии предприятия) BPWin позволяет переключиться на нотацию IDEF3 или DFD и создать

смешанную модель. Нотация DFD включает такие понятия, как внешняя ссылка и хранилище данных, что делает ее более удобной (по сравнению с IDEF0) для моделирования документооборота. Методология IDEF3 включает элемент «перекресток», что позволяет описать логику взаимодействия компонентов системы.

После построения функциональной модели можно построить модель данных, используя модель ERWin. Хотя процесс преобразования модели BPWin в модель данных плохо формализуется и поэтому полностью не автоматизируется, Computer Associations предлагает удобный инструмент для обеспечения построения модели данных на основе функциональной модели (механизм связи BPWin–ERWin). ERWin имеет два уровня представления модели – логический и физический. На логическом уровне данные не связаны с конкретной СУБД, поэтому могут быть наглядно представлены даже не для специалистов. Физический уровень данных – это, по существу, отображение системного каталога, который зависит от конкретной реализации СУБД. ERWin позволяет проводить процессы прямого и обратного проектирования БД. Это означает, что по модели данных можно сгенерировать схему БД или автоматически создать модель данных на основе информации системного каталога. Кроме того, ERWin интегрируется с популярными средствами разработки клиентской части – Power Builder, Visual Basic, Delphi, что позволяет автоматически генерировать код приложения, который полностью готов к компиляции и выполнению.

В заключение цикла проектирования МС рассмотрим еще одну возможность минимизации сложности МС и мехатронных модулей (ММ).

Синергетическая интеграция в мехатронных системах

В общем случае функциональная и структурная сложность технических систем (подсистем) в значительной степени определяется количеством соединяемых элементов, числом и интенсивностью их взаимодействия. Поэтому оснащение технических систем многофункциональными элементами, повышение уровня

интеграции ММ (подсистем) и снижение числа интерфейсов направлено на минимизацию сложности систем.

Функциональное представление МС и ММ с определенными входными и выходными переменными (модель типа «черный ящик») показано на рис. 8.2.



Рис. 8.2. Мехатронный модуль в виде «черного ящика»

Таким образом, основная функциональная задача мехатронной системы или мехатронного модуля заключается в преобразовании информации о программе движения в целенаправленное управляемое движение выходного звена.

Программа движения может быть задана управляющим компьютером как набор команд высокого уровня или, в случае дистанционного управления, человеком-оператором с помощью человеко-машинного интерфейса. Управляемое движение осуществляется механической подсистемой ММ, и его конечное звено взаимодействует с объектами внешней среды. Внешние воздействия, например силы резания при шлифовальных и фрезерных операциях, контактные силы и моменты при роботизированной сборке, должны эффективно отражаться мехатронным модулем в процессе движения. Информационная обратная связь необходима для оценки текущего состояния ММ как объекта управления и внешней среды в режиме реального времени.

Обозначенная основная функция не является единственной для мехатронных систем. Некоторые дополнительные функции, такие как реконфигурация системы, обмен сигналами и инфор-

мацией с другим технологическим оборудованием, самодиагностика, также должны быть реализованы для эффективной и надежной работы МС. Но именно выполнение функционального движения является основной мехатронной функцией, так как механическая подсистема взаимодействует с объектами работ и, таким образом, определяет поведение МС во внешней среде.

Рассмотренное представление мехатронного модуля в форме «черного ящика» (см. рис. 8.2) имеет два информационных входа (программа движения и информационная обратная связь), энергетический вход (реакция внешней среды) и механический выход (целенаправленное управляемое движение). Следовательно, в общем случае функциональная модель мехатронного модуля может быть определена как *информационно-механический преобразователь*.

Для физической реализации мехатронного информационно-механического преобразования необходим внешний энергетический источник. На современной стадии развития мехатроники для этой цели в основном используют электрические источники энергии. Введя соответствующие электроэнергетические преобразования, получаем следующую функциональную модель (*F-модель*) мехатронного модуля (рис. 8.3):

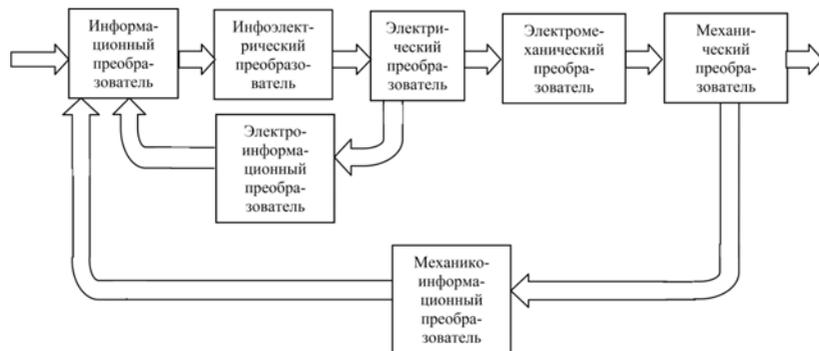


Рис. 8.3. Функциональная модель мехатронного модуля

Полученная *F-модель* ММ в общем случае содержит семь базовых функциональных преобразователей, связанных энергетическими и информационными потоками.

Отметим, что электрическая энергия является только промежуточной энергетической формой между входной информацией и выходным механическим движением. Следовательно, электрическая подсистема отнюдь не является единственно возможной для выполнения мехатронной функции, как это постулировано в приведенном определении мехатроники. Безусловно, и другие виды энергетических процессов могут быть использованы в мехатронных системах для промежуточных преобразований и должны рассматриваться как альтернативные варианты на этапе концептуального проектирования. Выбор разработчиком ММ физической природы промежуточного преобразователя определяется возможностями технической реализации, исходными требованиями и особенностями его применения. В современной инженерной практике широко применяются гидравлические, пневматические, химические и другие виды энергетических преобразователей.

Итак, в любом мехатронном модуле необходимо реализовать семь функциональных преобразований (см. рис. 8.3). Три из них назовем *моноэнергетическими* (информационный, электрический и механический преобразователи), где входные и выходные переменные имеют одну и ту же физическую природу. Остальные четыре являются *дуальными* (двойственными), так как в них входные и выходные переменные принадлежат различным физическим видам. К этой группе относят информационно-электрический и электромеханический преобразователи, расположенные в прямой цепи функциональной модели, и параллельные электроинформационный и механико-информационный преобразователи в цепи обратной связи.

Структурная модель мехатронного модуля должна отражать состав его элементов и связи между ними. В теории автоматического управления и электромеханике принято структурные модели графически представлять в виде блок-схем. Звенья обычно обозначают в виде прямоугольника с указанием входной и выходной переменных, а уравнения или характеристики записывают внутри него.

В качестве исходной структуры ММ рассмотрим традиционный электропривод с компьютерным управлением (S-модель –

рис. 8.4). Для дальнейшего анализа в представленной структурной схеме выделим управляющую и электромеханическую подсистемы. При конструировании мехатронных модулей особый интерес представляет исполнительная часть, входящая в состав электромеханической подсистемы.

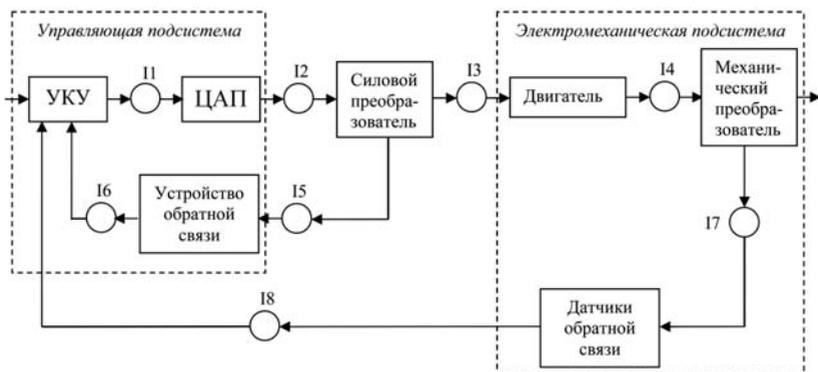


Рис. 8.4. Структурная модель электропривода с компьютерным управлением

Структурная модель электропривода (см. рис. 8.4) включает в себя следующие основные элементы:

- устройство компьютерного управления движением, функциональной задачей которого является информационное преобразование (обработка цифровых сигналов, цифровое регулирование, расчет управляющих воздействий, обмен данными с периферийными устройствами);
- цифро-аналоговый преобразователь (ЦАП), реализующий функцию информационно-электрического преобразования;
- силовой преобразователь, обычно состоящий из усилителя мощности, широтно-импульсного модулятора (ШИМ) и трехфазного инвертора (для асинхронных двигателей);
- управляемый электродвигатель (переменного или постоянного тока), который является электротехническим элементом привода;
- механический преобразователь, который реализует заданное управляемое движение и взаимодействует с внешними

объектами; в приводных модулях в качестве таких устройств применяют редукторы, вариаторы, либо непосредственно используют рабочий орган (например, в мехатронных модулях типа «мотор-шпиндель»);

- устройство обратной связи, которое используют для контроля текущих напряжений и токов в силовом преобразователе, а также управляющих функций (например, для организации контура регулирования момента, развиваемого приводом);

- датчики обратной связи по положению и скорости движения выходного звена механического устройства, выполняющие функции механико-информационного преобразования;

- интерфейсные устройства, обозначенные на блок-схеме как 11–18.

В зависимости от физической природы входных и выходных переменных интерфейсные блоки могут быть как механическими преобразователями движения, так и содержать электронные аппаратно-программные компоненты. Примерами механических интерфейсов являются передачи и трансмиссии, связывающие выходное механическое устройство с двигателем (интерфейс 14) и датчиками обратной связи (интерфейсы 17, 18).

Интерфейсные электронные устройства расположены на входах и выходах устройства компьютерного управления (УКУ) и предназначены для его сопряжения со следующими структурными элементами:

- с цифро-аналоговым преобразователем (встроенный интерфейс 11) и далее с силовым преобразователем модуля (интерфейс 12);

- с датчиками обратной связи (интерфейс 17), который в случае применения сенсоров с аналоговым выходным сигналом строится на основе аналого-цифрового преобразователя (АЦП);

- с устройствами обратной связи для контроля уровня электрических токов и напряжений в силовом преобразователе (для традиционного привода интерфейс 16 также использует стандартный АЦП).

В традиционной приводной технике интерфейсы являются сепаратными устройствами. Поэтому их проектирование, изгото-

товление и наладка становятся серьезной проблемой для разработчика привода, особенно когда требуется надежно соединять нестандартные и специализированные элементы различных производителей. Мехатронные структуры отличаются высокой степенью интегрированности элементов, причем эти решения закладываются уже на стадии проектирования модулей и машин.

Сравнивая функциональную модель мехатронного модуля (см. рис. 8.3) и структурную модель традиционного электропривода (см. рис. 8.4), можно сделать вывод о том, что суммарное количество основных и интерфейсных блоков в структуре электропривода значительно превышает число необходимых функциональных преобразователей. Другими словами, можно говорить о структурной избыточности традиционного электропривода. Наличие избыточных блоков приводит к снижению надежности и точности технической системы, ухудшению ее массогабаритных и стоимостных показателей. Поэтому целесообразно стремиться к сокращению количества отдельных структурных элементов (как основных, так и интерфейсных блоков) в системе. В идеальном для пользователя варианте мехатронный модуль (см. рис. 8.2), приняв на информационный вход программу движения, должен выполнить целенаправленное управляемое движение с заданными показателями качества. При этом все проблемы интеграции в модуле механических, электронных и управляющих устройств должны быть решены разработчиком для каждой стадии жизненного цикла – от проектирования системы до ее эксплуатации у конечного потребителя.

Суть синергетической интеграции состоит в объединении в единый модуль элементов различной физической природы при сохранении функционального преобразования, выполняемого данным модулем.

Синергетическая интеграция элементов при проектировании мехатронных модулей основана на трех базовых принципах:

- реализация заданных функциональных преобразований минимально возможным числом структурных и конструктивных блоков путем объединения двух и более элементов в единые многофункциональные модули;

– выбор интерфейсов в качестве локальных точек интеграции и исключение избыточных структурных блоков и интерфейсов как сепаратных элементов;

– перераспределение функциональной нагрузки в мехатронной системе от аппаратных блоков к интеллектуальным (электронным и компьютерным) компонентам.

Практическая реализация принципов синергетической интеграции при проектировании позволяет обеспечить основные преимущества мехатронных систем по сравнению с традиционными машинами и добиться качественно новых показателей, в первую очередь по компактности конструкции, скорости и точности движений. Снятие с аппаратной («железной») части системы функциональной нагрузки (прежде всего через упрощение механического узла) и ее перенос на управляющую и электронную подсистемы придает системе гибкость, делает ее способной к легкой реконфигурации под новые технологические задачи. Следует заметить, что интеграция предполагает не только аппаратное объединение элементов, но и организацию интегрированных информационных процессов в интеллектуальных модулях.

Синергетическую интеграцию в мехатронике осуществляют при проектировании двумя основными способами – функционально-структурной интеграцией (ФС-интеграция) и структурно-конструктивной интеграцией (СК-интеграция), которые объединяют в общий алгоритм проектирования мехатронных систем (см. рис. 8.2).

Функционально-структурная интеграция. Задачей этапа функционально-структурной интеграции является поиск мехатронных структур, реализующих заданные функциональные преобразования с помощью минимального количества структурных блоков. ФС-интеграция направлена на выбор проектных решений, которые обеспечивают исключение некоторых основных блоков, а значит, и смежных с ними интерфейсов из структуры системы.

Примеры мехатронных проектных решений, основанные на способе ФС-интеграции элементов, приведены в табл. 8.1. Пред-

ставленные решения основаны на совместном анализе функциональной модели мехатронного модуля (см. рис. 8.3) и структуры традиционного электропривода (см. рис. 8.4).

Первые два мехатронных решения относятся к электромеханической подсистеме модуля, следующие варианты интеграции можно реализовать в его управляющей подсистеме (см. рис. 8.4). При конструировании мехатронных модулей наибольшее внимание уделяют решениям, направленным на упрощение механической части модулей и связанных с ней блоков и интерфейсов, которые реализуют электромеханическое и механико-информационные функциональные преобразования.

Рассмотрим подробнее мехатронные решения по ФС-интеграции элементов из табл. 8.1.

Первый вариант предусматривает использование в обратной связи вместо двух отдельных датчиков положения и скорости только одного элемента – фотоимпульсного датчика (ФИД), который позволяет получать информацию и о угле поворота вала, и о скорости его вращения. При этом также важно, что ФИД выдает выходной сигнал в кодовой форме, что дает возможность вводить информацию в УКУ без дополнительного аналого-цифрового преобразователя (АЦП), что было необходимо для традиционных датчиков с аналоговым выходным сигналом (тахогенераторов, потенциометров и т. п.).

Различают два основных вида фотоимпульсных датчиков – *абсолютные* и *инкрементальные*. Абсолютные ФИД (*encoder*) дают информацию о величине перемещения (линейного или углового) подвижного элемента относительно фиксированного нулевого положения. Преимуществами абсолютного ФИД являются надежность измерения (даже при временном отключении питания информация датчиком не будет потеряна), высокая точность при больших скоростях движения, запоминание нулевого положения (это важно при необходимости управления реверсивными и аварийными движениями машин). Инкрементальный датчик дает информацию о направлении и величине перемещения в приращениях относительно исходного положения, которое он занимал до начала движения.

Таблица 8.1

ФС-интеграция элементов в мехатронном модуле

№ п/п	Мехатронное решение	Функциональное преобразование	Исключаемые сепаратные элементы	
			Основные блоки	Интерфейсы
1	Фотоимпульсный датчик обратной связи	Механико-информационное	Один датчик обратной связи	17, 18
2	Вентильный высокомоментный двигатель	Электромеханическое и механико-информационное	Механическое устройство, сепаратные датчики обратной связи	14, 17, 18
3	Интеллектуальный силовой преобразователь	Информационно-электрическое	Избыточное устройство обратной связи	15, 16
4	Управляющие контроллеры на базе блоков FPGA	Электроинформационное	Цифро-аналоговый преобразователь	11, 12

Интеллектуализацию ФИД обеспечивают встроенными микропроцессорами, которые выполняют следующие основные функции: кодирование информации датчика, обнаружение ошибок измерения, масштабирование сигнала и передачи текущего кода в контроллер движения по стандартному протоколу. Современная тенденция в создании ФИД заключается в объединении в одном сенсорном модуле конструктивных элементов (валов, подшипников), кодировочных дисков, фотоэлементов и микропроцессора.

Таким образом, использование ФИД позволяет исключить из структуры традиционного привода один датчик обратной связи с его интерфейсом (17), а также АЦП на входе УКУ (интерфейс 18).

Применение высокомоментного двигателя (ВМД) позволяет (второе решение в табл. 8.1) заменить исполнительную пару

«двигатель + преобразователь движения» на один приводной элемент «двигатель». Этот способ ФС-интеграции означает исключение механического устройства и избыточного интерфейса 14 из структуры привода.

Ниже перечислены основные преимущества мехатронных модулей с ВМД:

- снижение материалоемкости, компактность и модульность конструкции;
- повышенные точностные характеристики привода благодаря отсутствию зазоров, кинематических погрешностей, упругих деформаций звеньев и т. д.;
- исключение трения в механической трансмиссии, что позволяет избежать нелинейных динамических эффектов, особенно на ползучих скоростях.

Для определения положения полюсов на роторе двигателя в конструкцию вентильного ВМД встраивают датчик положения. В исполнительных приводах информацию с этого датчика могут использовать и как сигнал обратной связи. Следовательно, применение вентильных ВМД со встроенными ФИД позволяет упростить не только механическую часть модуля, но и цепь обратной связи, так как разработчику не требуется вводить в конструкцию модуля отдельные датчики положения и скорости.

ВМД могут быть как углового, так и линейного типа. До появления линейных двигателей традиционные электроприводы линейных перемещений включали в себя двигатель углового движения и механическую передачу для преобразования вращательного движения в поступательное (шарико-винтовую передачу, зубчатую рейку, ленточную передачу и т. п.). Основные преимущества мехатронных модулей на базе линейных двигателей по сравнению с традиционными приводами обусловлены исключением многоступенчатого преобразования движения, отсутствием характерных недостатков механических преобразователей (люфт, упругость, силы трения, высокая инерция). Это позволяет добиться повышения в несколько раз линейной скорости и ускорения, высокой точности реализации движения, повышенной статической и динамической жесткости привода.

В табл. 8.1 приведены также два примера применения способа ФС-интеграции к элементам управляющей подсистемы модуля.

В состав мехатронных модулей могут входить интеллектуальные силовые преобразователи (ИСП). Их строят на базе полупроводниковых приборов нового поколения. Типичными представителями этих приборов являются силовые полевые транзисторы (*MOSFET*), биполярные транзисторы с изолированным затвором (*IGBT*), запираемые тиристоры с полевым управлением (*MCT*). Новое поколение приборов отличается высоким быстродействием (например, для транзисторов *MOSFET* – 100 000 Гц), высокими значениями коммутируемых токов и напряжений (для *IGBT* предельная сила коммутируемого тока – до 1200 А, предельное коммутируемое напряжение – до 3 500 В). Особенность ИСП состоит в том, что они содержат встроенные блоки микроэлектроники, предназначенные для выполнения интеллектуальных функций – управление движением, защита в аварийных режимах и диагностика неисправностей. Использование ИСП в составе мехатронных модулей позволяет существенно снизить массогабаритные показатели силовых преобразователей, повысить их надежность при эксплуатации, улучшить технико-экономические показатели.

Использование контроллеров движения с блоками *FPGA* (*Field Programmable Gate Arrays*) позволяет исключить цифро-аналоговое преобразование сигналов при компьютерном управлении двигателем. На выходе блоков *FPGA* сразу формируется широтно-модулированный сигнал, который имеет цифровое представление. При этом они обладают уникальным сочетанием очень высокой производительности (скорость вычислений соизмерима с аппаратными компонентами) с возможностью программирования как обычные микропроцессорные устройства.

Обобщая рассмотренные примеры, интересно обратить внимание на то, что точками для ФС-интеграции являются структурные блоки, реализующие функциональные преобразования только дуального типа (см. третий столбец табл. 8.1). К этой группе относят информационно-электрический и электро-механический преобразователи, расположенные в прямой цепи

функциональной модели мехатронного модуля (см. рис. 8.3) и электроинформационный и механико-информационный преобразователи в цепи обратной связи.

Структурно-конструктивная интеграция (СК-интеграция) основана на анализе структурной модели мехатронного модуля, которая сформирована на этапе ФС-интеграции. Заданная структура модуля может быть реализована различными конструктивными решениями. СК-интеграция нацеливает разработчика мехатронных модулей на выбор проектных решений, которые обеспечивают исключение интерфейсов как сепаратных блоков путем встраивания их в отдельный корпус. При автоматизированном проектировании принятые решения представляют в виде конструктивной модели.

Методическим ключом при поиске вариантов СК-интеграции является рассмотрение интерфейсных блоков 11–18 в качестве локальных точек, где потенциально возможна СК-интеграция. Можно рекомендовать при проектировании опираться сразу на несколько точек интеграции.

Таблица 8.2

СК-интеграция элементов в мехатронном модуле

№ п/п	Мехатронные модули	Функциональное преобразование	Встраиваемые элементы	
			Основные блоки	Интерфейсы
1	Модуль движения	Электромеханическое и механическое	Двигатель, механическое устройство	14
2	Мехатронный модуль движения	Электромеханическое, механическое и механико-информационное	Двигатели, механическое устройство, датчик обратной связи	14, 17, 18
3	Интеллектуальный мехатронный модуль	Информационное, информационно-электрическое, электрическое, электромеханическое	Управляющий контроллер, силовой преобразователь, двигатель	11, 12, 13, 15, 16

Примеры мехатронных модулей, основанных на способе СК-интеграции элементов, приведены в табл. 8.2. Представленные решения базируются на анализе структурных моделей мехатронных модулей, разработанных на этапе функционально-структурной интеграции (см. рис. 8.4).

Интеграция элементов в мехатронных модулях является ведущей тенденцией при создании современных машин и систем, так как позволяет добиться качественно нового уровня по основным техническим показателям – скорости и точности движения, компактности конструкции и способности машины к быстрой реконфигурации. Практическое воплощение этой тенденции в машинах сегодняшнего дня зависит от эффективности взаимодействия конструктора, который выдвигает новые интеграционные идеи, и технолога, реализующего предложенные проектные решения в автоматизированных технологических процессах.

Специфика и сложность мехатронных модулей и систем заключается еще и в том, что их составляющие части (механическая, электронная и компьютерная) имеют различную физическую природу, а основные структурные элементы выпускаются зачастую предприятиями различных отраслей промышленности.

Разработка проектной и конструкторской документации

Процесс проектирования и конструирования мехатронной системы завершается разработкой проектной и конструкторской документации на систему, относится к *четвертому* этапу разработки системы.

Конструкторская документация (КД) подразделяется на проектную и рабочую.

Проектная конструкторская документация (ПКД) выполняется на различных стадиях реализации проекта в соответствии с техническим заданием (ТЗ), разработанным на предпроектной стадии (см. рис. 3.2), до разработки рабочей документации. В состав проектной документации входят: техническое предложение, эскизный и технический проекты. Состав и номенклатура документов,

входящих в техническое предложение, эскизный и технический проекты подробно изложены в лекции 3 (раздел 3.2).

Рабочая конструкторская документация (РКД) разрабатывается на основе технического задания или проектной конструкторской документации и предназначена для обеспечения монтажа, изготовления, контроля, приемки, эксплуатации и ремонта технических изделий (производственных процессов). Следует отметить, что выпуск и корректировка РКД обычно продолжается и на всех последующих этапах разработки и эксплуатации технической системы.

Четыре последние этапа (5, 6, 7, 8) (см. раздел 8.2) разработки мехатронных систем относятся к процессам подготовки, реализации и эксплуатации мехатронных систем (см. раздел 1.1, лекция 1). Сюда же можно отнести и процессы управления промышленными предприятиями на различных иерархических уровнях.

Основные положения этапа 5 «Разработка производства мехатронных систем (включая технологическую подготовку производства)» были рассмотрены в лекции 4 (раздел 4.1). Разработка больших производственных систем включает следующие виды работ:

- разработка производственных процессов, объединенных в производственную систему;
- разработка технологических процессов, входящих в состав производственных процессов;
- разработка технологических операций, на которые делится технологический процесс;
- разработка локальных систем управления отдельными процессами;
- разработка интегрированной информационной системы для автоматизации технологической подготовки и управления производственными системами и процессами;
- оптимизация технологических и производственных процессов;
- решение ряда инженерных задач.

Работы, связанные с выполнением 6 и 7 этапов, соответственно «Испытание» и «Доводка» мехатронных систем, в данном курсе лекций не рассматриваются.

Последний 8-й этап разработки мехатронных систем «Эксплуатация мехатронных систем» будет рассмотрен в лекции 9 в тесной увязке с этапом «Производство и реализация» жизненного цикла мехатронной системы (см. лекция 1, рис. 1.1). Основное внимание будет уделено автоматизированным системам управления промышленных предприятий, на которых реализуются производственные процессы.

Контрольные вопросы

1. Опишите традиционный алгоритм проектирования.
2. Сформируйте четыре уровня интеграции в мехатронных системах.
3. Разберите два направления проектных и конструкторских работ при проектировании и изготовлении мехатронных систем.
4. Опишите общую схему проектирования любой сложной системы.
5. Перечислите восемь этапов процесса разработки мехатронных систем (с выделением этапов, определяющих цикл проектирования).
6. Опишите инструментальные средства и методы информационных технологий, используемые при выполнении цикла проектирования.
7. Рассмотрите представление мехатронного модуля в форме «черного ящика».
8. Сформулируйте основную функциональную задачу мехатронной системы (или мехатронного модуля). Другие дополнительные функции мехатронных систем.
9. Опишите функциональную модель мехатронного модуля (F-модель).
10. Опишите структурную модель электропривода с компьютерным управлением (S-модель мехатронного модуля).
11. Сформулируйте суть синергетической интеграции элементов при проектировании мехатронных модулей. Рассмотрите три базовых принципа синергетической интеграции.

12. Сформулируйте суть функционально-структурной (ФС) и структурно-конструктивной (СК) синергетической интеграции.

13. Сформулируйте основную задачу этапа функционально-структурной интеграции (ФС-интеграция).

14. Рассмотрите мехатронные решения по функционально-структурной интеграции элементов из табл. 8.1.

15. Сформулируйте основную задачу этапа структурно-конструкторской интеграции (СК-интеграция).

16. Рассмотрите примеры мехатронных модулей, основанных на способе СК-интеграции (табл. 8.2).

Лекция 9

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ МЕХАТРОННЫМИ ПРОИЗВОДСТВАМИ [1, 3, 18]

Мехатронные производства имеют иерархическую многомодульную структуру и соответствующие этой структуре автоматизированные системы управления (АСУ). Среди АСУ различают автоматизированные системы управления предприятием (АСУП) и автоматизированные системы управления технологическими процессами (АСУТП).

9.1. Автоматизация управления предприятиями (АСУП)

В большинстве случаев АСУП охватывают уровни от предприятия до цеха.

Функции АСУП на промышленных предприятиях выполняют следующие автоматизированные системы (на различных иерархических уровнях) (см. лекцию 1, раздел 1.1):

ERP – Enterprise Resource Planning (планирование и управление предприятием);

MRP (MRP-II) – Manufacturing (Material) Requirement Planning (планирование производственных (материальных) ресурсов);

MES – Manufacturing Execution System (производственная исполнительная система);

DRP – Distribution Resource Planning (управление маркетинговыми мероприятиями и обработкой заказов);

FRP – Finance Requirements Planning (финансовый учет и планирование);

PM – Project Management (управление проектами).

Имеется ряд автоматизированных систем (АС):

CRM – Customer Relationship Management (управление продажами), SCM – Supply Chain Management (управление цепочками поставок), S&SM – Sales and Service Management (управление продажами и обслуживанием). Эти АС могут быть отнесены как к АСУП, так и к логистическим системам, которые будут рассмотрены ниже.

Исторически первоначально был разработан стандарт MRP. Он появился в конце 60-х гг. Стандарт MRP включал в себя только планирование потребностей в материалах по замкнутому циклу. Затем стандарт MRP был расширен до MRP-II. Это позволило планировать все производственные ресурсы предприятия на основе данных, полученных от поставщиков и потребителей. Но этого оказалось недостаточно, и появился стандарт ERP, объединяющий возможности MRP и MRP-II, а также возможности стандартов DRP и FRP. Однако ERP-системы имеют явный крен в сторону автоматизации и стандартизации финансовых и учетно-хозяйственных функций производственных процессов и не обеспечивают интегрированного решения задач технологической подготовки производствами с его оперативным управлением, ограничиваясь только стратегическим планированием, а также между уровнем ERP и уровнем технологической подготовки и оперативного управления производством. В результате оказывается не охваченным информационными технологиями целый класс жизненно важных для предприятия производственных процессов.

Опыт внедрения ERP показал, что при создании действительно интегрированной системы управления предприятием в первую очередь необходимо решить в едином информационном пространстве (ЕИП) взаимоувязанные задачи автоматизации технологической подготовки и оперативного управления производством.

В настоящее время создание интегрированных информационных систем управления промышленными предприятиями реализуется на базе CALS-технологий, которые в гражданской

промышленности называются PLM-системами (PLM – Product Lifecycle Management – управление жизненным циклом изделий).

Автоматизированные информационные системы (АИС) класса PLM включают в свой состав PDM-системы в качестве электронного архива и технологического документооборота (workflow).

В [18] приведены зарубежные и российские компании-разработчики АИС-систем класса PLM. Среди российских компаний-разработчиков следует выделить: «Аскон», «Топ Системы», НПП «Интермех» (г. Минск), «Уфимский государственный авиационный технический университет» (УГАТУ). Все эти компании разрабатывают АИС-системы класса PLM с программным обеспечением (ПО), позволяющим интегрировать в состав АИС-систем автоматизированные системы (АС), САПР: CAD, CAM, CAE.

На повестку дня поставлена задача определения базовых информационных технологий ИАИС (интегрированная автоматизированная информационная система) для алгоритмического и программного обеспечения функций CAD/CAM/CAE/PDM/FRP/MRP/MES-систем в ЕИП предприятия.

В настоящее время системами корпоративного управления и учета для планирования материальных ресурсов (MRP и MRP-II), управления технической информацией (TIM), электронной почтой, маршрутизацией документов при коллективной работе и т. д. являются: R|3, BAAN, AVALON9 (CIM) MANMAN/X, MMI, SOCAP, TRITON, а также STAFFWARE, SCALA и Lotus Notes. Самой крупной интегрированной системой этого класса является R|3.

9.2. Автоматизация управления технологическими процессами (АСУТП)

Системы АСУТП охватывают уровень от цеха и ниже. Однако в АСУТП могут быть включены и межцеховые связи, если управление единым производственным процессом выполняется в нескольких цехах. Эти системы часто называют системами промышленной автоматизации, или встроенными системами (Embedded Systems).

В состав АСУТП входят автоматизированные системы:

- SCADA – Supervisory Control and Data Acquisition (диспетчерское управление производственными процессами);
- CNC – Computer Numerical Control (компьютерное численное управление).

Система SCADA на верхнем иерархическом уровне АСУТП выполняет диспетчерские функции:

- сбор и обработка данных о состоянии оборудования;
- контроль протекания технологических процессов;
- выполняет роль инструментальной системы разработки ПО для встроенного в состав АСУТП оборудования компьютерной автоматизации;
- управление и регистрация аварийных сигналов (алармов);
- связь с корпоративной информационной сетью.

SCADA-системы состоят из терминальных компонентов, диспетчерских пунктов и каналов связи. Связь с контроллерами и приложениями в SCADA-системах обычно осуществляется посредством технологий: DDE – Dynamic Data Exchange (обмен данными), OLE – Object Linking and Embedding Technology (связывание и внедрение объектов), ODBC – Open Database Connectivity (открытый доступ к базам данных). В качестве каналов связи используют последовательные промышленные шины Profibus, CANbus, Foundation Fieldbus и др.

Разработка программ для программируемых контроллеров выполняется на языках C/C++, VBA – Visual basic для приложений или оригинальных языках, разработанных для конкретных систем. Ряд языков стандартизован и представлен в международном стандарте IEC 1131–3.

Известные SCADA-системы:

- Citect (компания Ci Technology), работающая в среде Windows;
- Trace Mode (компания AdAstra), операционные системы QNX, OS9, Windows;
- LabView SCADA (компания National Instruments).

Функциями систем MES являются анализ производственных процессов, их оптимизация, управление ресурсами и расходом

материалов, анализ простоя оборудования, диагностика и предупреждение поломок оборудования, контроль и управление качеством продукции, формирование отчетов о производстве для передачи на уровень ERP.

Функции MES близки по некоторым выполняемым функциям к системам ERP, PDM, SCM, S&SM и отличаются от них именно оперативностью, принятием решений в реальном времени, причем большое значение придается оптимизации этих решений с учетом текущей информации о состоянии оборудования и процессов.

В настоящее время наблюдается интеграция АСУП и АСУТП, ранее развивающихся автономно. Использование информации непосредственно от технологических процессов позволяет более рационально планировать производство и управлять предприятием за счет использования на этих уровнях общих программных средств, баз данных, сетей Industrial Ethernet и др.

На уровне управления технологическим оборудованием (запуск, тестирование, выключение оборудования, сигнализация о неисправностях, выработка управляющих воздействий для программно управляемого оборудования) используются программируемые контроллеры с системой CNC. Эти контроллеры обычно встраиваются в технологическое оборудование.

Техническое обеспечение АСУТП представлено персональными ЭВМ (в промышленном исполнении), контроллерами CNC, микрокомпьютерами PLC, распределенными по контролируемым участкам производства, связанными друг с другом с помощью промышленных шин.

9.3. Информационные логистические системы

Логистические задачи занимают значительное место среди задач, решаемых при организации управления сложными производственными системами.

Исторически в начале логистику связывали с управлением процедурами движения сырья и готовой продукции, их склады-

рованием и движением к месту потребления. Логистика формировалась как «наука об организации совместной деятельности менеджеров различных подразделений предприятия, а также группы предприятий по эффективному продвижению продукции по цепи «закупка сырья – производство продукции – сбыт – распределение» на основе интеграции и координации операций, процедур и функций, выполняемых в рамках данного процесса с целью минимизации общих затрат ресурсов». По существу, логистика выполняла функции подсистемы АСУП.

В 1974 г. международный конгресс по логистике предложил более широкое определение логистики «как науки о планировании, управлении и контроле за движением материальных, информационных и финансовых ресурсов в различных системах».

Затем появился термин «информационная логистика», под которым понимают организацию и использование систем информационного обеспечения производственно-хозяйственных процессов на предприятии. Объект информационной логистики – информационные потоки, сопровождающие материальные потоки.

Информационная логистика наиболее актуальна для автоматизированных систем проектирования (САПР) и управления, оперирующих информационными ресурсами, например в виде потоков документов (в различных формах), порождаемых материальными потоками и сопровождающих материальные потоки. Информационная логистика охватывает все виды деятельности, связанные с планированием и управлением процессами, нацеленными на обеспечение предприятия релевантной информацией.

В процессе проектирования мехатронных изделий или систем вся информация по проекту хранится в банке изделия, но возникает проблема с ее доставкой на рабочее место исполнителя. Возможностей СУБД для этого уже недостаточно. Более эффективные методы управления информацией оформлены в виде международных стандартов группы STEP (ISO 10303) CALS-системы. Такие методы составляют суть современной информационной логистики. Вместо логистической системы нередко

используются родственные ей системы управления разработками (проектами), например PDM-система управления проектными данными [10].

Создание логистической системы для САПР, обеспечивающей разработку или модернизацию сложных технических систем, начинается с описания предполагаемого изделия (входящих в его состав узлов (подсистем) и связей между ними) на специальном алгоритмическом языке EXPRESS, имеющимся в ISO 10303 (см. лекцию 7). При этом производится декомпозиция изделия – детальная функциональная и формируется структура информации, передаваемой по всем предусмотренным типам связи.

Полученная выше система пока обеспечивает только управление потоками информации. В ней не содержится никаких расчетов. Все расчеты, конструирование, анализ полученной информации, ее визуализация и прочая работа, необходимая для создания изделия, выполняется в отдельных модулях. Модули создаются на базе существующих САПР. Модули должны быть оформлены в виде самостоятельных программ со стандартными интерфейсами, предусмотренными в логистической системе. К числу таких модулей относятся системы ERP, SCM, CRM, PDM и др.

Вся созданная система размещается в локальной сети, например типа клиент-сервер. В этом случае все версии банка изделия и банка знаний размещаются на достаточно мощном сервере (типа Сгау-3). В этом случае число клиентов в сети может измеряться сотнями.

Затем разрабатывается схема процесса проектирования изделия и начинается проектирование изделия.

В [10] более подробно описывается технология проектирования и последующая работа над изделием.

Логическим решением разработки информационных логистических систем является создание интегрированной автоматизированной системы управления (ИАИСУ), обеспечивающей информационное взаимодействие всех участников проекта.

9.4. Автоматизированные системы делопроизводства

В связи с переходом машиностроительной промышленности от серийного к мелкосерийному производству значительно возросла номенклатура выпускаемых предприятиями наукоемких изделий, что привело к многократному росту объема технической информации и документооборота.

На этапах конструкторской и технологической подготовки производства ведется коллективная работа с большим объемом документации, которая представлена в разных форматах (текстовые документы, чертежи, спецификации, отчеты, таблицы и т. д.) и разрабатываются в разных автоматизированных информационных системах (АИС) с использованием различных методов проектирования и типов сопровождающих их данных (твердотельные 3D модели, схемы печатных плат, программы для ЧПУ и т. д.).

Существует достаточно большое количество критериев классификации автоматизированных систем делопроизводства (АСД) по целевому назначению, приложению, масштабам:

- системы управления документами (СУД);
- системы управления документооборотом (СДО);
- инструментальные среды делопроизводства;
- специализированные и комплексные системы;
- локальные и распределенные системы;
- заказные и тиражируемые системы;
- фактографические и полнотекстовые системы;
- системы управления техническими данными (TDM);
- офисные системы делопроизводства;
- системы управления знаниями.

Отличия TDM от офисных АСД заключается в большом количестве чертежной документации и в тесной взаимосвязи с САПР.

Системы управления документами (СУД) предназначены для обеспечения санкционированного доступа к документам.

Систему управления документооборотом служат для управления деловыми процессами прохождения и обработки документов в соответствующих подразделениях и службах предприятия.

К системам управления знаниями в области делопроизводства относятся системы, выполняющие функции, характерные для интеллектуальных систем (классификация документов, тематический отбор документов, аналитическая обработка данных, моделирование деловых процессов).

Инструментальные среды в системах делопроизводства служат для формирования АСД, адаптированных к условиям конкретных предприятий и фирм. Ряд других задач, решаемых АСД, описан в [3, 18].

Основными компонентами СУД и СДО являются:

- электронный архив;
- система маршрутизации документов;
- контроль исполнения;
- модули сопровождения с прикладными программами;
- средства ввода информации.

В крупных АСД предусматривается распределенное хранение с доступом к документам в режимах off-line и on-line.

Современные системы делопроизводства являются распределенными, имеющими архитектуру клиент-сервер. На серверной стороне применяются серверы баз данных полнотекстовых документов, электронной почты, приложений SQL- и Web-серверы. На клиентской стороне – рабочие места пользователей.

К широкоизвестным СДО и АСД относятся Lotus Notes, Docs Open и др. примерами TDM служат Search или T-Flex Docs (на небольшое число пользователей) и системы документооборота, используемые в PDM Optegra или iMAN (для большого числа пользователей).

Контрольные вопросы

1. Перечислите автоматизированные системы, выполняющие функции АСУП на промышленных предприятиях.
2. Опишите стандарт MRP, входящий в АСУП.
3. Опишите стандарт MRP-II, входящий в АСУП.
4. Перечислите самые крупные интегрированные автоматизированные информационные системы (ИАИС).

5. Перечислите автоматизированные системы, выполняющие функции АСУП.
 6. Укажите автоматизированные системы, выполняющие функции АСУТП.
 7. Перечислите функции системы SCADA.
 8. Перечислите известные SCADA-системы.
 9. Опишите функции MES-систем.
 10. В чем выражается интеграция АСУП и АСУТП?
 11. Опишите историю развития логистических систем.
 12. Дайте определение понятию «информационная логистика».
 13. Укажите область применения информационной логистики.
 14. Опишите порядок создания логистической системы для САПР.
 15. Перечислите критерии классификации автоматизированных систем делопроизводства (АСД).
- Опишите основные компоненты систем управления документами (СУД) и систем управления документооборотом (СДО).

ЗАКЛЮЧЕНИЕ

Итак, мы рассмотрели курс лекций, посвященных изучению основ дисциплины «Информационная поддержка мехатронных комплексов и производств», являющейся базовой составляющей более общей дисциплины «Проектирование мехатронных систем».

Данный курс лекций составляет в совокупности основу знаний в области информационного обеспечения разработки сложных (мехатронных) систем различного назначения и позволяет оценить роль информационного обеспечения при современном подходе к проектированию любых сложных технических систем.

Авторы не обещают, что прочитав эту книгу, студент сможет непосредственно притупить к разработке мехатронных систем (в лекциях основное внимание уделено только общим подходам к решению этой проблемы). Но авторы уверены, что студент будет подготовлен к дальнейшему самостоятельному углубленному изучению специальной технической литературы по всем аспектам, связанным с разработкой сложных технических и организационных систем.

Желаем вам больших успехов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Норенков И. П. К. Информационная поддержка наукоемких изделий. CALS-технологии / И. П. Норенков, П. К. Кузьмик. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2002. – 320 с.
2. Моисеев Н. Н. Математические задачи системного анализа / Н. Н. Моисеев. – М. : Наука. Главная редакция физико-математической литературы, 1981. – 488 с.
3. Норенков И. П. Основы автоматизированного проектирования : учеб. для вузов / И. П. Норенков. – 2-е изд., перераб. и доп. – М. : Изд-во МГТУ им. Н. Э. Баумана. 2002. – 336 с.
4. Готлиб Б. М. Технология автоматизированного машиностроения : учеб. пособие : в 2 т. / Б. М. Готлиб, А. А. Вакалюк, М. Б. Готлиб. – Т. 2. Автоматизированные высокоэнергетические технологические процессы точного машиностроения. – Екатеринбург : УрГУПС. – 2012. – 230 с.
5. Юревич Е. И. Основы робототехники : учеб. пособие / Е. И. Юревич. – 3-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2010. – 308 с.
6. Черемных С. В. структурный анализ систем: IDEF-технологии / С. В. Черемных, И. О. Семенов, В. С. Ручкин. – М. : Финансы и статистика, 2001. – 208 с.
7. Леопенков А. В. Самоучитель UML / А. В. Леопенков. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2006. – 432 с.
8. Бенькович Е. С. Практическое моделирование динамических систем / Е. С. Бенькович, Ю. Б. Колесов, Ю. Б. Сениченков. – СПб. : БХВ-Петербург, 2002. – 464 с.
9. Ли Кунву. Основы САПР (CAD/CAM/CAE) / Ли Кунву. – СПб. : Питер, 2004. – 560 с.
10. Тунаков А. П. САПР газотурбинных двигателей : учеб. пособие / А. П. Тунаков, И. А. Кривошеев, Д. А. Ахмедзянов. – Уфа : УГАТУ, 2005. – 272 с.
11. Кривошеев И. А. Формализация процесса проектирования и доводки двигателей с использованием CASE-технологии / И. А. Кривошеев. – М. : Изд-во «МАИ», 2008. – 128 с.

12. Лукинов А. П. Проектирование мехатронных и робототехнических устройств : учеб. пособие / А. П. Лукинов. – СПб. : Изд-во «Лань», 2012. – 608 с.
13. Подураев Ю. В. Мехатроника: основы, методы, применение : учеб. пособие для студентов вузов / Ю. В. Подураев. – М. : Машиностроение, 2006. – 256 с.
14. Егоров О. Д. Мехатронные модули. Расчет и проектирование : учеб. пособие / О. Д. Егоров, Ю. В. Подураев. – М. : МГТУ «Станкин», 2004. – 360 с.
15. Марка Д. Методология структурного анализа и проектирования / Д. Марка, К. МакГоуэн. – М. : Метатехнология, 1992. – 239 с.
16. Файзрахманов Р. А. Структурно-функциональный подход к проектированию информационных технологий и автоматизированных систем с использованием CASE-средств / Р. А. Файзрахманов, К. А. Селезнев. – Пермь : ПГТУ, 2005. – 245 с.
17. Кривошеев И. А. Компьютерное моделирование в инновационном проектировании авиационных двигателей / И. А. Кривошеев, С. Г. Селиванов. – М. : Машиностроение, 2010. – 330 с.
18. Кульга К. С. Модели и методы создания интегрированной информационной системы для автоматизации технологической подготовки и управления авиационным и машиностроительным производством : монография / К. С. Кульга, И. А. Кривошеев. – М. : Машиностроение, 2011. – 377 с.
19. Кривошеев И. А. Общая структура автоматизированной разработки авиационных двигателей и энергоустановок / И. А. Кривошеев. – М. : Машиностроение, 2009. – 265 с.

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



УТВЕРЖДАЮ

И. о. проректора по учебно-методической работе _____ В.В. Зубов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ И ЗАДАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

ФТД.В.01 ТЕХНОЛОГИИ ИНТЕЛЛЕКТУАЛЬНОГО ТРУДА

Специальность

21.05.04 Горное дело

Специализация

Мехатроника и робототехника промышленных комплексов

Автор: Полянок О.В., к.пс.н., доцент

Одобрена на заседании кафедры

Управления персоналом

(название кафедры)

Зав. кафедрой

(подпись)

Беляева Е.А.

(Фамилия И.О.)

Протокол № 1 от 06.09.2024

(Дата)

Рассмотрена методической комиссией

Горно-механического факультета

(название факультета)

Председатель

(подпись)

Осипов П.А.

(Фамилия И.О.)

Протокол № 1 от 18.10.2024

(Дата)

Екатеринбург

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ОСНОВНЫЕ КАТЕГОРИИ ДИСЦИПЛИНЫ.....	6
САМООРГАНИЗАЦИЯ РАБОТЫ С ЛИТЕРАТУРОЙ.....	8
ПОДГОТОВКА К ПРАКТИКО-ОРИЕНТИРОВАННЫМ ЗАДАНИЯМ.....	12
ПРАКТИКО-ОРИЕНТИРОВАННЫЕ ЗАДАНИЯ.....	13
ПОДГОТОВКА К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	28

ВВЕДЕНИЕ

Самостоятельная работа в высшем учебном заведении - это часть учебного процесса, метод обучения, прием учебно-познавательной деятельности, комплексная целевая стандартизованная учебная деятельность с запланированными видом, типом, формами контроля.

Самостоятельная работа представляет собой плановую деятельность обучающихся по поручению и под методическим руководством преподавателя.

Целью самостоятельной работы студентов является закрепление тех знаний, которые они получили на аудиторных занятиях, а также способствование развитию у студентов творческих навыков, инициативы, умению организовать свое время.

Самостоятельная работа реализует следующие задачи:

- предполагает освоение курса дисциплины;
- помогает освоению навыков учебной и научной работы;
- способствует осознанию ответственности процесса познания;
- способствует углублению и пополнению знаний студентов, освоению ими навыков и умений;
- формирует интерес к познавательным действиям, освоению методов и приемов познавательного процесса,
- создает условия для творческой и научной деятельности обучающихся;
- способствует развитию у студентов таких личных качеств, как целеустремленность, заинтересованность, исследование нового.

Самостоятельная работа обучающегося выполняет следующие функции:

- развивающую (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей студентов);
- информационно-обучающую (учебная деятельность студентов на аудиторных занятиях, неподкрепленная самостоятельной работой, становится мало результативной);
- ориентирующую и стимулирующую (процессу обучения придается ускорение и мотивация);
- воспитательную (формируются и развиваются профессиональные качества бакалавра и гражданина);
- исследовательскую (новый уровень профессионально-творческого мышления).

Организация самостоятельной работы студентов должна опираться на определенные требования, а, именно:

- сложность осваиваемых знаний должна соответствовать уровню развития студентов;
- стандартизация заданий в соответствии с логической системой курса дисциплины;
- объем задания должен соответствовать уровню студента;
- задания должны быть адаптированными к уровню студентов.

Содержание самостоятельной работы студентов представляет собой, с одной стороны, совокупность теоретических и практических учебных заданий, которые должен выполнить студент в процессе обучения, объект его деятельности; с другой стороны - это способ деятельности студента по выполнению соответствующего теоретического или практического учебного задания.

Свое внешнее выражение содержание самостоятельной работы студентов находит во всех организационных формах аудиторной и внеаудиторной деятельности, в ходе самостоятельного выполнения различных заданий.

Функциональное предназначение самостоятельной работы студентов в процессе лекций, практических занятий по овладению специальными знаниями заключается в самостоятельном прочтении, просмотре, прослушивании, наблюдении, конспектировании, осмыслении, запоминании и воспроизведении определенной информации. Цель и планирование самостоятельной работы студента определяет преподаватель. Вся информация осуществляется на основе ее воспроизведения.

Так как самостоятельная работа тесно связана с учебным процессом, ее необходимо рассматривать в двух аспектах:

1. аудиторная самостоятельная работа – практические занятия;
2. внеаудиторная самостоятельная работа – подготовка к практическим занятиям (в т.ч. подготовка к практико-ориентированным заданиям и др.).

Основные формы организации самостоятельной работы студентов определяются следующими параметрами:

- содержание учебной дисциплины;
- уровень образования и степень подготовленности студентов;
- необходимость упорядочения нагрузки студентов при самостоятельной работе.

Таким образом, самостоятельная работа студентов является важнейшей составной частью процесса обучения.

Методические указания по организации самостоятельной работы и задания для обучающихся по дисциплине *«Технологии интеллектуального труда»* обращают внимание студента на главное, существенное в изучаемой дисциплине, помогают выработать умение анализировать явления и факты, связывать теоретические положения с практикой, а также облегчают подготовку к сдаче *зачета*.

Настоящие методические указания позволят студентам самостоятельно овладеть фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю подготовки, опытом творческой и исследовательской деятельности, и направлены на формирование компетенций, предусмотренных учебным планом поданному профилю.

Видами самостоятельной работы обучающихся по дисциплине *«Технологии интеллектуального труда»* являются:

- самостоятельное изучение тем курса (в т.ч. рассмотрение основных категорий дисциплины, работа с литературой);

- подготовка к практическим (семинарским) занятиям (в т.ч. ответы на вопросы для самопроверки, подготовка к выполнению практико-ориентированных заданий);
- подготовка к зачету.

В методических указаниях представлены материалы для самостоятельной работы и рекомендации по организации отдельных её видов.

ОСНОВНЫЕ КАТЕГОРИИ ДИСЦИПЛИНЫ

Тема 1. Особенности информационных технологий для людей с ограниченными возможностями.

Информационные технологии
Универсальный дизайн
Адаптивные технологии

Тема 2. Тифлотехнические средства/ Сурдотехнические средства/ Адаптивная компьютерная техника (Материал изучается по подгруппам в зависимости от вида ограничений здоровья обучающихся)

Брайлевский дисплей
Брайлевский принтер
Телевизионное увеличивающее устройство
Читающая машина
Экранные лупы
Синтезаторы речи
Ассистивные тифлотехнические средства
Ассистивные сурдотехнические средства
Адаптированная компьютерная техника
Ассистивные технические средства

Тема 3. Дистанционные образовательные технологии

Дистанционные образовательные технологии
Информационные объекты

Тема 4. Интеллектуальный труд и его значение в жизни общества

Система образования
Образовательная среда вуза
Интеллектуальный труд
Интеллектуальный ресурс
Интеллектуальный продукт

Тема 5. Развитие интеллекта – основа эффективной познавательной деятельности

Личностный компонент
Мотивационно-потребностный компонент
Интеллектуальный компонент
Организационно-деятельностный компонент
Гигиенический компонент
Эстетический компонент
Общеучебные умения
Саморегуляция

Тема 6. Самообразование и самостоятельная работа студента – ведущая форма умственного труда.

Самообразование

Самостоятельная работа студентов

Технологии интеллектуальной работы

Технологии групповых обсуждений

Тема 7. Технологии работы с информацией студентов с ОВЗ и инвалидов

Традиционные источники информации

Технологии работы с текстами

Технологии поиска, фиксирования, переработки информации

Справочно-поисковый аппарат книги

Техника быстрого чтения

Реферирование

Редактирование

Технология конспектирования

Методы и приемы скоростного конспектирования

Тема 8. Организация научно-исследовательской работы

Доклад

Реферат

Курсовая работа

Выпускная квалификационная работа

Техника подготовки работы

Методика работы над содержанием Презентация

Тема 9. Тайм-менеджмент

Время

Планирования времени

Приемы оптимизации распределения времени

САМООРГАНИЗАЦИЯ РАБОТЫ С ЛИТЕРАТУРОЙ

Самостоятельное изучение тем курса осуществляется на основе списка рекомендуемой литературы к дисциплине. При работе с книгой необходимо научиться правильно ее читать, вести записи. Самостоятельная работа с учебными и научными изданиями профессиональной и общекультурной тематики – это важнейшее условие формирования научного способа познания.

Основные приемы работы с литературой можно свести к следующим:

- составить перечень книг, с которыми следует познакомиться;
- перечень должен быть систематизированным;
- обязательно выписывать все выходные данные по каждой книге (при написании курсовых и выпускных квалификационных работ это позволит экономить время);

- определить, какие книги (или какие главы книг) следует прочитать более внимательно, а какие – просто просмотреть;

- при составлении перечней литературы следует посоветоваться с преподавателями, которые помогут сориентироваться, на что стоит обратить большее внимание, а на что вообще не стоит тратить время;

- все прочитанные монографии, учебники и научные статьи следует конспектировать, но это не означает, что надо конспектировать «все подряд»: можно выписывать кратко основные идеи автора и иногда приводить наиболее яркие и показательные цитаты (с указанием страниц);

- если книга – собственная, то допускается делать на полях книги краткие пометки или же в конце книги, на пустых страницах просто сделать свой «предметный указатель», где отмечаются наиболее интересные мысли и обязательно указываются страницы в тексте автора;

- следует выработать способность «воспринимать» сложные тексты; для этого лучший прием – научиться «читать медленно», когда понятно каждое прочитанное слово (а если слово незнакомое, то либо с помощью словаря, либо с помощью преподавателя обязательно его узнать). Таким образом, чтение текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации.

От того, насколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия. Грамотная работа с книгой, особенно если речь идет о научной литературе, предполагает соблюдение ряда правил, для овладения которыми необходимо настойчиво учиться. Это серьезный, кропотливый труд. Прежде всего, при такой работе невозможен формальный, поверхностный подход. Не механическое заучивание, не простое накопление цитат, выдержек, а сознательное усвоение прочитанного, осмысление его, стремление дойти до сути – вот главное правило. Другое правило – соблюдение при работе над книгой определенной последовательности. Вначале следует ознакомиться с оглавлением,

содержанием предисловия или введения. Это дает общую ориентировку, представление о структуре и вопросах, которые рассматриваются в книге.

Следующий этап – чтение. Первый раз целесообразно прочитать книгу с начала до конца, чтобы получить о ней цельное представление. При повторном чтении происходит постепенное глубокое осмысление каждой главы, критического материала и позитивного изложения; выделение основных идей, системы аргументов, наиболее ярких примеров и т.д. Непременным правилом чтения должно быть выяснение незнакомых слов, терминов, выражений, неизвестных имен, названий. Студентам с этой целью рекомендуется заводить специальные тетради или блокноты. Важная роль в связи с этим принадлежит библиографической подготовке студентов. Она включает в себя умение активно, быстро пользоваться научным аппаратом книги, справочными изданиями, каталогами, умение вести поиск необходимой информации, обрабатывать и систематизировать ее.

Выделяют четыре основные установки в чтении текста:

- информационно-поисковая (задача – найти, выделить искомую информацию);

- усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить, как сами сведения, излагаемые автором, так и всю логику его рассуждений);

- аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему);

- творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

С наличием различных установок обращения к тексту связано существование и нескольких видов чтения:

- библиографическое – просматривание карточек каталога, рекомендательных списков, сводных списков журналов и статей за год и т.п.;

- просмотровое – используется для поиска материалов, содержащих нужную информацию, обычно к нему прибегают сразу после работы со списками литературы и каталогами, в результате такого просмотра читатель устанавливает, какие из источников будут использованы в дальнейшей работе;

- ознакомительное – подразумевает сплошное, достаточно подробное прочтение отобранных статей, глав, отдельных страниц; цель – познакомиться с характером информации, узнать, какие вопросы вынесены автором на рассмотрение, провести сортировку материала;

- изучающее – предполагает доскональное освоение материала; в ходе такого чтения проявляется доверие читателя к автору, готовность принять изложенную информацию, реализуется установка на предельно полное понимание материала;

- аналитико-критическое и творческое чтение – два вида чтения близкие между собой тем, что участвуют в решении исследовательских задач.

Первый из них предполагает направленный критический анализ, как самой информации, так и способов ее получения и подачи автором; второе – поиск тех суждений, фактов, по которым, или, в связи с которыми, читатель считает нужным высказать собственные мысли.

Из всех рассмотренных видов чтения основным для студентов является изучающее – именно оно позволяет в работе с учебной и научной литературой накапливать знания в различных областях. Вот почему именно этот вид чтения в рамках образовательной деятельности должен быть освоен в первую очередь. Кроме того, при овладении данным видом чтения формируются основные приемы, повышающие эффективность работы с текстом. Научная методика работы с литературой предусматривает также ведение записи прочитанного. Это позволяет привести в систему знания, полученные при чтении, сосредоточить внимание на главных положениях, зафиксировать, закрепить их в памяти, а при необходимости вновь обратиться к ним.

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения.

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала.

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала.

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора.

Конспектирование – краткое и последовательное изложение содержания прочитанного. Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Как правильно составлять конспект? Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта. Выделите главное, составьте план, представляющий собой перечень заголовков, подзаголовков, вопросов, последовательно раскрываемых затем в конспекте. Это первый элемент конспекта. Вторым элементом конспекта являются тезисы. Тезис - это кратко сформулированное положение. Для лучшего усвоения и запоминания материала следует записывать тезисы своими словами. Тезисы, выдвигаемые в конспекте, нужно доказывать. Поэтому третий элемент конспекта - основные доводы, доказывающие истинность рассматриваемого тезиса. В конспекте могут быть положения и примеры. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли. При оформлении

конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Конспектирование – наиболее сложный этап работы. Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы. Конспект ускоряет повторение материала, экономит время при повторном, после определенного перерыва, обращении к уже знакомой работе. Учитывая индивидуальные особенности каждого студента, можно дать лишь некоторые, наиболее оправдавшие себя общие правила, с которыми преподаватель и обязан познакомить студентов:

1. Главное в конспекте не объем, а содержание. В нем должны быть отражены основные принципиальные положения источника, то новое, что внес его автор, основные методологические положения работы. Умение излагать мысли автора сжато, кратко и собственными словами приходит с опытом и знаниями. Но их накоплению помогает соблюдение одного важного правила – не торопиться записывать при первом же чтении, вносить в конспект лишь то, что стало ясным.

2. Форма ведения конспекта может быть самой разнообразной, она может изменяться, совершенствоваться. Но начинаться конспект всегда должен с указания полного наименования работы, фамилии автора, года и места издания; цитаты берутся в кавычки с обязательной ссылкой на страницу книги.

3. Конспект не должен быть «слепым», безликим, состоящим из сплошного текста. Особо важные места, яркие примеры выделяются цветным подчеркиванием, взятием в рамочку, оттенением, пометками на полях специальными знаками, чтобы можно было быстро найти нужное положение. Дополнительные материалы из других источников можно давать на полях, где записываются свои суждения, мысли, появившиеся уже после составления конспекта.

ПОДГОТОВКА К ПРАКТИКО-ОРИЕНТИРОВАННЫМ ЗАДАНИЯМ

Практико-ориентированные задания выступают средством формирования у студентов системы интегрированных умений и навыков, необходимых для освоения профессиональных компетенций. Это могут быть ситуации, требующие применения умений и навыков, специфичных для соответствующего профиля обучения (знания содержания предмета), ситуации, требующие организации деятельности, выбора её оптимальной структуры личностно-ориентированных ситуаций (нахождение нестандартного способа решения).

Кроме этого, они выступают средством формирования у студентов умений определять, разрабатывать и применять оптимальные методы решения профессиональных задач. Они строятся на основе ситуаций, возникающих на различных уровнях осуществления практики и формулируются в виде производственных поручений (заданий).

Под практико-ориентированными заданиями понимают задачи из окружающей действительности, связанные с формированием практических навыков, необходимых в повседневной жизни, в том числе с использованием элементов производственных процессов.

Цель практико-ориентированных заданий – приобретение умений и навыков практической деятельности по изучаемой дисциплине.

Задачи практико-ориентированных заданий:

- закрепление, углубление, расширение и детализация знаний студентов при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- обучение приемам решения практических задач;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Важными отличительными особенностями практико-ориентированных задания от стандартных задач (предметных, межпредметных, прикладных) являются:

- значимость (познавательная, профессиональная, общекультурная, социальная) получаемого результата, что обеспечивает познавательную мотивацию обучающегося;
- условие задания сформулировано как сюжет, ситуация или проблема, для разрешения которой необходимо использовать знания из разных разделов основного предмета, из другого предмета или из жизни, на которые нет явного указания в тексте задания;

- информация и данные в задании могут быть представлены в различной форме (рисунок, таблица, схема, диаграмма, график и т.д.), что потребует распознавания объектов;

- указание (явное или неявное) области применения результата, полученного при решении задания.

Кроме выделенных четырех характеристик, практико-ориентированные задания имеют следующие:

1. по структуре эти задания – нестандартные, т.е. в структуре задания не все его компоненты полностью определены;

2. наличие избыточных, недостающих или противоречивых данных в условии задания, что приводит к объемной формулировке условия;

3. наличие нескольких способов решения (различная степень рациональности), причем данные способы могут быть неизвестны учащимся, и их требуется сконструировать.

При выполнении практико-ориентированных заданий следует руководствоваться следующими общими рекомендациями:

- для выполнения практико-ориентированного задания необходимо внимательно прочитать задание, повторить лекционный материал по соответствующей теме, изучить рекомендуемую литературу, в т.ч. дополнительную;

- выполнение практико-ориентированного задания включает постановку задачи, выбор способа решения задания, разработку алгоритма практических действий, программы, рекомендаций, сценария и т. п.;

- если практико-ориентированное задание выдается по вариантам, то получить номер варианта исходных данных у преподавателя; если нет вариантов, то нужно подобрать исходные данные самостоятельно, используя различные источники информации;

- для выполнения практико-ориентированного задания может использоваться метод малых групп. Работа в малых группах предполагает решение определенных образовательных задач в рамках небольших групп с последующим обсуждением полученных результатов. Этот метод развивает навыки сотрудничества, достижения компромиссного решения, аналитические способности.

ПРАКТИКО-ОРИЕНТИРОВАННЫЕ ЗАДАНИЯ

1. В соответствии с опросником «Саморегуляция» (ОС) (модификация методики А.К. Осницкого) оцените свои качества, возможности, отношение к деятельности в протоколе (132 высказывания) по 4-х бальной шкале: 4 балла – да; 3 балла – пожалуй да; 2 балла – пожалуй нет; 1 балл – нет.

Текст опросника

1. Способен за дело приниматься без напоминаний.
2. Планирует, организует свои дела и работу.
3. Умеет выполнить порученное задание.
4. Хорошо анализирует условия.
5. Учитывает возможные трудности.
6. Умеет отделять главное от второстепенного.
7. Чаще всего избирает верный путь решения задачи.
8. Правильно планирует свои занятия и работу.
9. Пытается решить задачи разными способами.
10. Сам справляется с возникающими трудностями.
11. Редко ошибается, умеет оценить правильность действий.
12. Быстро обнаруживает свои ошибки.
13. Быстро находит новый способ решения.
14. Быстро исправляет ошибки.
15. Не повторяет ранее сделанных ошибок.
16. Продумывает свои дела и поступки.
17. Хорошо справляется и с трудными заданиям.
18. Справляется с заданиями без посторонней помощи.
19. Любит порядок.
20. Заранее знает, что будет делать.
21. Аккуратен и последователен.
22. Продумывает, все до мелочей.
23. Ошибается чаще из-за того, что смысл задания целом не понят, хотя все детали продуманы.
24. Старателен, хотя часто не выполняет заданий.
25. Долго готовится, прежде чем приступить к делу.
26. Избегает риска.
27. Сначала обдумывает, потом делает.
28. Решения принимает без колебаний.
29. Уверенный в себе.
30. Действует решительно, настойчив.
31. Предприимчивый, решительный.
32. Активный.
33. Ведущий.
34. Реализует почти все, что планирует.
35. Начатое дело доводит до конца.
36. Предпочитает действовать, а не обсуждать.

37. Обдумывает свои дела и поступки.
38. Анализирует свои ошибки и неудачи.
39. Планирует дела, рассчитывает свои силы.
40. Прислушивается к замечаниям.
41. Редко повторяет одну и ту же ошибку.
42. Знает о своих недостатках.
43. Сделает задание на совесть.
44. Как всегда сделает на отлично.
45. Для него важно качество, а не отметка.
46. Всегда проверяет правильность работы.
47. Старается довести дело до конца.
48. Стирается добиться лучших результатов.
49. Действует самостоятельно, мало советуясь с другими.
50. Предпочитает справляться с трудностями сам.
51. Может принять не зависимое от других решение.
52. Любит перемену в занятиях.
53. Легко переключается с одной работы на другую.
54. Хорошо ориентируется в новых условиях.
55. Аккуратен.
56. Внимателен.
57. Усидчив.
58. С неудачами и ошибками обычно справляется.
59. Неудачи активизируют его.
60. Старается разобраться в причинах неудач.
61. Умеет мобилизовать усилия.
62. Взвешивает все «за» и «против».
63. Старается придерживаться правил.
64. Всегда считается с мнением других.
65. Его нетрудно убедить в чем-то.
66. Прислушивается к замечаниям.
67. Нужно напоминать о том, что необходимо закончить дело.
68. Не планирует, мало организует свои дела, и работу.
69. Не выполняет заданий оттого, что отвлекается.
70. Условия анализирует плохо.
71. Не учитывает возможных трудностей.
72. Не умеет отделять главное от второстепенного.
73. Пути решения выбирает не лучшие.
74. Не умеет планировать работу и занятия.
75. Не пытается решать задачи разными способами.
76. Не может справиться с трудностями без помощи других.
77. Часто допускает ошибки в работе, часто их повторяет.
78. С трудом находит ошибки в своей работе.
79. С трудом находит новые способы решения.
80. С большим трудом и долго исправляет ошибки.

81. Повторяет одни и те же ошибки.
82. Часто поступает необдуманно, импульсивно.
83. С трудными заданиями справляется плохо.
84. Не справляется с заданием без напоминаний и помощи.
85. Не любит порядок.
86. Часто не знает заранее, что ему предстоит делать.
87. Непоследователен и неаккуратен.
88. Ограничивается лишь общими сведениями, общим впечатлением.
89. Ошибается чаще из-за того, что не продуманы мелочи, детали.
90. Не очень старателен, но задания выполняет.
91. Приступает к делу без подготовки.
92. Часто рискует, ищет приключений.
93. Сначала сделает, лотом подумает.
94. Решения принимает после раздумий и колебаний.
95. Часто сомневается в своих силах.
96. Нерешителен, небольшие помехи уже останавливают его.
97. Нерешительный.
98. Вялый, безучастный.
99. Ведомый.
100. Задумывает много, а делает мало.
101. Редко, когда начатое дело доводит до конца.
102. Предпочитает обсуждать, а не действовать.
103. Действует без раздумий, «с ходу».
104. Не анализирует ошибок.
105. Не планирует почти ничего, не рассчитывает своих сил.
106. Не прислушивается к замечаниям.
107. Часто повторяет одну и ту же ошибку.
108. Не хочет знать и исправлять свои недостатки.
109. Сделает «спустя рукава».
110. Сделает как получится.
111. Сделает из-за угрозы получения плохой оценки.
112. Не проверяет правильность результатов своих действий.
113. Часто бросает работу, не доделав ее.
114. Результат неважен – лишь бы поскорее закончить работу.
115. О его трудностях и делах знают почти все.
116. Всегда надеется на друзей, на их помощь.
117. Действует по принципу: как все, так и я!
118. Любит однообразные занятия.
119. С трудом переключается с одной работы на другую.
120. Плохо ориентируется в новых условиях.
121. Неаккуратен.
122. Невнимателен.
123. Неусидчив.
124. Ошибку может исправить, если его успокоить.

125. Неудачи быстро сбивают с толку.
126. Равнодушен к причинам неудач.
127. С трудом мобилизуется на выполнение задания.
128. Поступает необдуманно, импульсивно.
129. Не придерживается правил.
130. Не считается с мнением окружающих.
131. Его трудно убедить в чем-либо.
132. Не прислушивается к замечаниям.

Ключ для обработки и интерпретации данных

В тесте оценивается 132 характеристики саморегуляции. Они разбиты на тройки.

Всего 22 пары противоположных характеристик.

1. Целеполагание - 23. Неустойчивость целей.
2. Моделирование условий - 24. Отсутствие анализа условий.
3. Программирование действий - 25. Спонтанность действий.
4. Оценивание результатов - 26. Ошибки в работе.
5. Коррекции результатов и способ» действий - 27. Повторные ошибки.
6. Обеспеченность регуляции в целом - 28. Импульсивность.
7. Упорядоченность деятельности - 29. Непоследовательность, неаккуратность.
8. Детализация регуляции действий - 30. Поверхностность.
9. Осторожность в действиях - 31. Необдуманность, рискованность.
10. Уверенность в действиях - 32. Неуверенность в своих силах.
11. Инициативность в действиях - 33. Нерешительность.
12. Практическая реализуемость намерений - 34. Незавершенность дел.
13. Осознанность действий - 35. Действия наобум.
14. Критичность в делах и поступках -36. Равнодушие к недостаткам.
15. Ориентированность на оценочный балл -37. Попустительство.
16. Ответственность в делах и поступках - 38. Безответственность в делах.
17. Автономность - 39. Зависимость в действиях.
18. Гибкость, пластичность в действиях - 40. Инертность в работе.
19. Вовлечение полезных привычек в регуляцию действий - 41. «Плохиш».
20. Практичность, устойчивость в регуляции действий - 42. Равнодушие к ошибкам, неудачам.
21. Оптимальность (адекватность) регуляции усилий - 43. Отсутствие последовательности.
22. Податливость воспитательным воздействиям - 44. Самодостаточность.

Необходимо найти сумму в каждой из троек характеристик и сопоставить ее с их противоположностью.

4-6 баллов - слабое проявление характеристики.

7-9 баллов - ситуативное проявление.

10-12 баллов - выраженность характеристики.

Бланк для ответов

ФИ _____
 Пол _____ Возраст (дата рождения) _____ Гр. _____ Дата _____ № _____

Шкала ответов

4 – да; 3 – пожалуй да; 2 – пожалуй нет; 1 – нет.

№			S		№	
1	1			23	67	
	2				68	
	3				69	
2	4			24	70	
	5				71	
	6				72	
3	7			25	73	
	8				74	
	9				75	
4	10			26	76	
	11				77	
	12				78	
5	13			27	79	
	14				80	
	15				81	
6	16			28	82	
	17				83	
	18				84	
7	19			29	85	
	20				86	
	21				87	
8	22			30	88	
	23				89	
	24				90	
9	25			31	91	
	26				92	
	27				93	

S

10	28		32	94	
	29			95	
	30			96	
11	31		33	97	
	32			98	
	33			99	
12	34		34	100	
	35			101	
	36			102	
13	37		35	103	
	38			104	
	39			105	
14	40		36	106	
	41			107	
	42			108	
15	43		37	109	
	44			ΠΟ	
	45			111	
16	46		38	112	
	47			113	
	48			114	
17	49		39	115	
	50			116	
	51			117	
18	52		40	118	
	53			119	
	54			120	
19	55		41	121	
	56			122	
	57			123	
20	58		42	124	
	59			125	
	60			126	

21	61		43	127	
	62			128	
	63			129	
22	64		44	130	
	65			131	
	66			132	

Качественные характеристики саморегуляции

№	Качества саморегуляции	Содержательные характеристики саморегуляции	№	Качества саморегуляции	Содержательные характеристики саморегуляции
1	Целеполагание	За дело приниматься без напоминаний, планирует, организует свои дела и работу. Задания и поручения выполняет.	23	Неустойчивость целей	Не планирует, мало организует свою работу. Нужно напоминать о том, что необходимо закончить дело. Отвлекается.
2	Моделирование условий	Анализирует условия предстоящей деятельности, возможные трудности. Выделяет главное.	24	Отсутствие анализа условий	Не умеет отделять главное от второстепенного. Не предвидит ход дел, возможные трудности.
3	Программирование действий	Правильно планирует свои занятия и работу, избирает верный путь решения задачи.	25	Спонтанность действий	Не умеет планировать работу в занятия, затрудняется в выборе путей решения задач.
4	Оценивание результатов	Редко ошибается, умеет оценить правильность действий. Быстро обнаруживает свои ошибки.	26	Ошибки в работе	Часто допускает ошибки в работе, часто их повторяет. Не находит ошибок в своей работе.
5	Коррекция результатов и способов действий	Быстро находит новый способ решения. Быстро исправляет ошибки.	27	Повторные ошибки	С трудом находит новые способы решения. Повторяет одни и те же ошибки.
6	Обеспеченность регуляции в целом	Продумывает свои дела и поступки. Справляется с заданиями без посторонней помощи.	28	Импульсивность	Часто поступает необдуманно, импульсивно. С трудными заданиями справляется плохо.

№	Качества саморегуляции	Содержательные характеристики саморегуляции	№	Качества саморегуляции	Содержательные характеристики саморегуляции
		щи.			
7	Упорядоченность деятельности	Любит порядок. Аккуратен и последователен.	29	Непоследовательность	Часто не знает заранее, что ему предстоит делать, непоследователен и неаккуратен.
8	Детализация регуляции действий	Продумывает, все до мелочей. Ошибается чаще из-за того, что смысл задания целом не понят, хотя все детали продуманы.	30	Поверхностность	Ограничивается лишь общими сведениями, общим впечатлением. Ошибается чаще из-за того, что не продуманы мелочи, детали.
9	Осторожность в действиях	Долго обдумывает и готовится, прежде чем приступить к делу. Избегает риска.	31	Необдуманность, рискованность	Приступает к делу без подготовки. Сначала делает, потом подумает.
10	Уверенность в действиях	Уверенный в себе. Решения принимает без колебаний. Решителен. Настойчив.	32	Неуверенность в своих силах	Решения принимает после колебаний. Сомневается в своих силах. Нерешителен.
11	Инициативен в действиях.	Предприимчивый, решительный. Активный. Ведущий.	33	Нерешительность	Нерешительный. Вялый, безучастный. Ведомый.
12	Практическая реализуемость намерений	Реализует почти все, что планирует. Начатое дело доводит до конца.	34	Незавершенность дел	Редко, когда начатое дело доводит до конца. Предпочитает обсуждать, а не действовать.
13	Осознанность действий	Обдумывает, планирует свои дела и поступки. Анализирует свои ошибки и неудачи.	35	Действия наобум	Действует без раздумий, «с ходу», не рассчитывает своих сил.
14	Критичность в делах и поступках	Знает о своих недостатках. Редко повторяет ошибки. Прислушивается к замечаниям.	36	Равнодушие к недостаткам	Часто повторяет одну и ту же ошибку. Не хочет знать и исправлять свои недостатки.

№	Качества саморегуляции	Содержательные характеристики саморегуляции	№	Качества саморегуляции	Содержательные характеристики саморегуляции
15	Ориентированность на оценочный балл	Сделает задание на совесть. Для него важно качество, а не отметка.	37	Попустительство	Делает все «спустя рукава», как получится. Делает из-за угрозы плохой оценки.
16	Ответственность в делах и поступках	Гарантирует доведение дел до конца. Всегда проверяет правильность работы.	38	Безответственность в делах	Не проверяет результатов своих действий. Часто бросает работу, не доделав до конца.
17	Автономность	Действует и принимает самостоятельные решения. Предпочитает сам справляться с трудностями.	39	Зависимость в действиях	Всегда надеется на друзей, на их помощь.
18	Гибкость, пластичность в действиях	Легко переключается с одной работы на другую. Хорошо ориентируется в новых условиях.	40	Инертность в работе	Любит однообразные занятия. С трудом переключается с одной работы на другую.
19	Вовлечение полезных привычек в регуляцию действий	Аккуратен. Внимателен. Усидчив.	41	«Плохиш»	Неаккуратен. Невнимателен. Неусидчив.
20	Практичность, устойчивость в регуляции действий	Справляется с неудачами и ошибками. Неудачи активизируют его. Старается разобраться в их причинах.	42	Равнодушие к ошибкам, неудачам	Неудачи быстро сбивают с толку. Равнодушен к их причинам.
21	Оптимальность (адекватность) регуляции усилий	Взвешивает все «за» и «против». Умеет мобилизовать усилия.	43	Отсутствие последовательности	Поступает необдуманно. С трудом мобилизуется на выполнение задания.
22	Податливость воспитательным воздействиям	Всегда считается с мнением других. Прислушивается к замечаниям.	44	Самодостаточность	Не считается с мнением окружающих. Не прислушивается к замечаниям.

Задание: На основе самодиагностики саморегуляции сформулируйте рекомендации по саморегуляции.

2. Выберите научную статью по своей специальности и напишите к ней аннотацию, реферат, конспект, рецензию.

Методические указания

АННОТАЦИЯ (от лат. *annotatio* - замечание, пометка) – это краткая характеристика статьи, рукописи, книги, в которой обозначены тема, проблематика и назначение издания, а также содержатся сведения об авторе и элементы оценки книги.

Перед текстом аннотации даются выходные данные (автор, название, место и время издания). Эти данные можно включить в первую часть аннотации.

Аннотация обычно состоит из двух частей. В первой части формулируется основная тема книги, статьи; во второй части перечисляются (называются) основные положения. Говоря схематично, аннотация на книгу (прежде всего научную или учебную) отвечает на вопросы о чем? из каких частей? как? для кого? Это ее основные, стандартные смысловые элементы. Каждый из них имеет свои языковые средства выражения.

Аннотация на книгу помещается на оборотной стороне ее титульного листа и служит (наряду с ее названием и оглавлением) источником информации о содержании работы. Познакомившись с аннотацией, читатель решает, насколько книга может быть ему нужна. Кроме того, умение аннотировать прочитанную литературу помогает овладению навыками реферирования.

Языковые стереотипы, с помощью которых оформляется каждая смысловая часть аннотации:

1. Характеристика содержания текста:

В статье (книге) рассматривается...; Статья посвящена...; В статье даются...; Автор останавливается на следующих вопросах...; Автор затрагивает проблемы...; Цель автора – объяснить (раскрыть)...; Автор ставит своей целью проанализировать...;

2. Композиция работы:

Книга состоит из ... глав (частей)...; Статья делится на ... части; В книге выделяются ... главы.

3. Назначение текста:

Статья предназначена (для кого; рекомендуется кому)...; Сборник рассчитан...; Предназначается широкому кругу читателей...; Для студентов, аспирантов...; Книга заинтересует...

РЕФЕРАТ (от лат. *referre*- докладывать, сообщать) – это композиционно организованное, обобщенное изложение содержания источника информации (статьи, ряда статей, монографии и др.). Реферат отвечает на вопрос: «Какая информация содержится в первоисточнике, что излагается в нем?»

Реферат состоит из трех частей: общая характеристика текста (выходные данные, формулировка темы); описание основного содержания; выводы референта. Изложение одной работы обычно содержит указание на тему и композицию реферируемой работы, перечень ее основных положений с приведением аргументации, реже - описание методики и проведение эксперимента, результатов и выводов исследования. Такой реферат называется про-

стым информационным. Студенты в российских вузах пишут рефераты обычно на определенные темы. Для написания таких тематических рефератов может быть необходимо привлечение более чем одного источника, по крайней мере двух научных работ. В этом случае реферат является не только информационным, но и обзорным.

Реферирование представляет собой интеллектуальный творческий процесс, включающий осмысление текста, аналитико-синтетическое преобразование информации и создание нового текста. Реферат не должен превращаться в «ползанье» по тексту. Цель реферирования – создать «текст о тексте». Реферат – это не конспект, разбавленный «скрепами» типа *далее автор отмечает...* Обильное цитирование превращает реферат в конспект. При чтении научного труда важно понять его построение, выделить смысловые части (они будут основой для плана), обратить внимание на типичные языковые средства (словосочетания, вводные конструкции), характерные для каждой части. В реферате должны быть раскрыты проблемы и основные положения работы, приведены доказательства этих положений и указаны выводы, к которым пришел автор. Реферат может содержать оценочные элементы, например: *нельзя не согласиться, автор удачно иллюстрирует* и др. Обратите внимание, что в аннотации проблемы научного труда лишь обозначаются, а в реферате – раскрываются.

Список конструкций для реферативного изложения:

Предлагаемая вниманию читателей статья (книга, монография) представляет собой детальное (общее) изложение вопросов...; Рассматриваемая статья посвящена теме (проблеме, вопросу...);

Актуальность рассматриваемой проблемы, по словам автора, определяется тем, что...; Тема статьи (вопросы, рассматриваемые в статье) представляет большой интерес...; В начале статьи автор дает обоснование актуальности темы (проблемы, вопроса, идеи); Затем дается характеристика целей и задач исследования (статьи);

Рассматриваемая статья состоит из двух (трех) частей...; Автор дает определение (сравнительную характеристику, обзор, анализ)...; Затем автор останавливается на таких проблемах, как...; Автор подробно останавливается на истории возникновения (зарождения, появления, становления)...; Автор подробно (кратко) описывает (классифицирует, характеризует) факты...; Автор доказывает справедливость (опровергает что-либо)...; Автор приводит доказательства справедливости своей точки зрения...; В статье дается обобщение..., приводятся хорошо аргументированные доказательства...;

В заключение автор говорит о том, что...; Несомненный интерес представляют выводы автора о том, что...; Наиболее важными из выводов автора представляются следующие...; Изложенные (рассмотренные) в статье вопросы (проблемы) представляют интерес не только для..., но и для...

КОНСПЕКТИРОВАНИЕ – письменная фиксация основных положений читаемого или воспринимаемого на слух текста. При конспектировании происходит свертывание, компрессия первичного текста.

КОНСПЕКТ- это краткое, но связное и последовательное изложение значимого содержания статьи, лекции, главы книги, учебника, брошюры. Запись-конспект позволяет восстановить, развернуть с необходимой полнотой исходную информацию, поэтому при конспектировании надо отбирать новый и важный материал и выстраивать его в соответствии с логикой изложения. В конспект заносят основные (существенные) положения, а также фактический материал (цифры, цитаты, примеры). В конспекте последующая мысль должна вытекать из предыдущей (как в плане и в тезисах). Части конспекта должны быть связаны внутренней логикой, поэтому важно отразить в конспекте главную мысль каждого абзаца. Содержание абзаца (главная мысль) может быть передано словами автора статьи (возможно сокращение высказывания) или может быть изложено своими словами более обобщенно. При конспектировании пользуются и тем и другим приемом, но важно передать самые главные положения автора без малейшего искажения смысла.

Различают несколько видов конспектов в зависимости от степени свернутости первичного текста, от формы представления основной информации:

1. конспект-план;
2. конспект-схема;
3. текстуальный конспект.

Подготовка конспекта включает следующие этапы:

1. Вся информация, относящаяся к одной теме, собирается в один блок – так выделяются смысловые части.
2. В каждой смысловой части формулируется тема в опоре на ключевые слова и фразы.
3. В каждой части выделяется главная и дополнительная по отношению к теме информация.
4. Главная информация фиксируется в конспекте в разных формах: в виде тезисов (кратко сформулированных основных положений статьи, доклада), выписок (текстуальный конспект), в виде вопросов, выявляющих суть проблемы, в виде назывных предложений (конспект-план и конспект-схема).
5. Дополнительная информация приводится при необходимости.

РЕЦЕНЗИЯ - это письменный критический разбор какого-либо произведения, предполагающий, во-первых, комментирование основных положений (толкование авторской мысли; собственное дополнение к мысли, высказанной автором; выражение своего отношения к постановке проблемы и т.п.); во-вторых, обобщенную аргументированную оценку, в третьих, выводы о значимости работы.

В отличие от рецензии ОТЗЫВ дает самую общую характеристику работы без подробного анализа, но содержит практические рекомендации: анализируемый текст может быть принят к работе в издательстве или на соискание ученой степени.

Типовой план для написания рецензии и отзывов:

1. Предмет анализа: *В работе автора...; В рецензируемой работе...; В предмете анализа...*

2. Актуальность темы: Работа посвящена актуальной теме...; Актуальность темы обусловлена...; Актуальность темы не вызывает сомнений (вполне очевидна)...

3. Формулировка основного тезиса: Центральным вопросом работы, где автор добился наиболее существенных (заметных, ощутимых) результатов, является...; В работе обоснованно на первый план выдвигается вопрос о...

4. Краткое содержание работы.

5. Общая оценка: Оценивая работу в целом...; Таким образом, рассматриваемая работа...; Автор проявил умение разбираться в...; систематизировал материал и обобщил его...; Безусловной заслугой автора является новый методический подход (предложенная классификация, некоторые уточнения существующих понятий); Автор, безусловно, углубляет наше представление об исследуемом явлении, вскрывает новые его черты...

6. Недостатки, недочеты: Вместе с тем вызывает сомнение тезис о том...; К недостаткам (недочетам) работы следует отнести допущенные автором длины в изложении (недостаточную ясность при изложении)...; Работа построена нерационально, следовало бы сократить...; Существенным недостатком работы является...; Отмеченные недостатки носят чисто локальный характер и не влияют на конечные результаты работы...; Отмеченные недочеты работы не снижают ее высокого уровня, их скорее можно считать пожеланиями к дальнейшей работе автора...; Упомянутые недостатки связаны не столько с..., сколько с...

7. Выводы: Представляется, что в целом работа... имеет важное значение...; Работа может быть оценена положительно, а ее автор заслуживает...; Работа заслуживает высокой (положительной, отличной) оценки...; Работа удовлетворяет всем требованиям..., а ее автор, безусловно, имеет (определенное, законное, заслуженное, безусловное) право...

Задание

а) Выберите научную статью по своей специальности и напишите к ней аннотацию, реферат, конспект, рецензию.

3. Проанализируйте отрывок из студенческой курсовой работы, посвященной проблеме связи заголовка и текста. Соответствует ли язык сочинения нормам научного стиля? На основании анализа проведите правку текста:

Заголовок, будучи неотъемлемой частью газетных публикаций, определяет лицо всей газеты. Сталкиваясь с тем или иным периодическим изданием, читатель получает первую информацию о нем именно из заголовков. На примере газеты «Спорт – экспресс» за апрель – май 1994 г. я рассмотрю связь: заголовок – текст, ведь, как говорится в народной мудрости «встречают по одежке, а провожают – по уму». Но даже при наличии прекрасной одежки (заглавий) и величайшего ума (самих материалов) стилистическая концепция газеты будет не полной, если будет отсутствовать продуманная и логичная связь между содержанием и заголовком. Итак, стараясь выбрать наиболее продуманные заглавия,

я попытаюсь проследить за тем, по какому принципу строится связь между содержанием и заголовком самой популярной спортивной газеты России «Спорт – экспресс». А к тому же я остановлюсь и на классификации заголовков по типу их связей с газетным текстом вообще.

ПОДГОТОВКА К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

При подготовке к *зачету* по дисциплине «*Технологии интеллектуального труда*» обучающемуся рекомендуется:

1. повторить пройденный материал и ответить на вопросы, используя конспект и материалы лекций. Если по каким-либо вопросам у студента недостаточно информации в лекционных материалах, то необходимо получить информацию из раздаточных материалов и/или учебников (литературы), рекомендованных для изучения дисциплины «*Технологии интеллектуального труда*».

Целесообразно также дополнить конспект лекций наиболее существенными и важными тезисами для рассматриваемого вопроса;

2. при изучении основных и дополнительных источников информации в рамках выполнения заданий на *зачете* особое внимание необходимо уделять схемам, рисункам, графикам и другим иллюстрациям, так как подобные графические материалы, как правило, в наглядной форме отражают главное содержание изучаемого вопроса;

3. при изучении основных и дополнительных источников информации в рамках выполнения заданий на *зачете* (в случаях, когда отсутствует иллюстративный материал) особое внимание необходимо обращать на наличие в тексте словосочетаний вида «во-первых», «во-вторых» и т.д., а также дефисов и перечислений (цифровых или буквенных), так как эти признаки, как правило, позволяют структурировать ответ на предложенное задание.

Подобную текстовую структуризацию материала слушатель может трансформировать в рисунки, схемы и т. п. для более краткого, наглядного и удобного восприятия (иллюстрации целесообразно отразить в конспекте лекций – это позволит оперативно и быстро найти, в случае необходимости, соответствующую информацию);

4. следует также обращать внимание при изучении материала для подготовки к *зачету* на словосочетания вида «таким образом», «подводя итог сказанному» и т.п., так как это признаки выражения главных мыслей и выводов по изучаемому вопросу (пункту, разделу). В отдельных случаях выводы по теме (разделу, главе) позволяют полностью построить (восстановить, воссоздать) ответ на поставленный вопрос (задание), так как содержат в себе основные мысли и тезисы для ответа.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ОСНОВНЫЕ КАТЕГОРИИ ДИСЦИПЛИНЫ.....	6
САМООРГАНИЗАЦИЯ РАБОТЫ С ЛИТЕРАТУРОЙ.....	8
ПОДГОТОВКА К ПРАКТИКО-ОРИЕНТИРОВАННЫМ ЗАДАНИЯМ.....	12
ПРАКТИКО-ОРИЕНТИРОВАННЫЕ ЗАДАНИЯ.....	14
ПОДГОТОВКА РЕФЕРАТА.....	36
ПОДГОТОВКА К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	45

ВВЕДЕНИЕ

Самостоятельная работа в высшем учебном заведении - это часть учебного процесса, метод обучения, прием учебно-познавательной деятельности, комплексная целевая стандартизованная учебная деятельность с запланированными видом, типом, формами контроля.

Самостоятельная работа представляет собой плановую деятельность обучающихся по поручению и под методическим руководством преподавателя.

Целью самостоятельной работы студентов является закрепление тех знаний, которые они получили на аудиторных занятиях, а также способствование развитию у студентов творческих навыков, инициативы, умению организовать свое время.

Самостоятельная работа реализует следующие задачи:

- предполагает освоение курса дисциплины;
- помогает освоению навыков учебной и научной работы;
- способствует осознанию ответственности процесса познания;
- способствует углублению и пополнению знаний студентов, освоению ими навыков и умений;
- формирует интерес к познавательным действиям, освоению методов и приемов познавательного процесса,
- создает условия для творческой и научной деятельности обучающихся;
- способствует развитию у студентов таких личных качеств, как целеустремленность, заинтересованность, исследование нового.

Самостоятельная работа обучающегося выполняет следующие функции:

- развивающую (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей студентов);
- информационно-обучающую (учебная деятельность студентов на аудиторных занятиях, неподкрепленная самостоятельной работой, становится мало результативной);
- ориентирующую и стимулирующую (процессу обучения придается ускорение и мотивация);
- воспитательную (формируются и развиваются профессиональные качества бакалавра и гражданина);
- исследовательскую (новый уровень профессионально-творческого мышления).

Организация самостоятельной работы студентов должна опираться на определенные требования, а, именно:

- сложность осваиваемых знаний должна соответствовать уровню развития студентов;
- стандартизация заданий в соответствии с логической системой курса дисциплины;
- объем задания должен соответствовать уровню студента;
- задания должны быть адаптированными к уровню студентов.

Содержание самостоятельной работы студентов представляет собой, с одной стороны, совокупность теоретических и практических учебных заданий, которые должен выполнить студент в процессе обучения, объект его деятельности; с другой стороны - это способ деятельности студента по выполнению соответствующего теоретического или практического учебного задания.

Свое внешнее выражение содержание самостоятельной работы студентов находит во всех организационных формах аудиторной и внеаудиторной деятельности, в ходе самостоятельного выполнения различных заданий.

Функциональное предназначение самостоятельной работы студентов в процессе лекций, практических занятий по овладению специальными знаниями заключается в самостоятельном прочтении, просмотре, прослушивании, наблюдении, конспектировании, осмыслении, запоминании и воспроизведении определенной информации. Цель и планирование самостоятельной работы студента определяет преподаватель. Вся информация осуществляется на основе ее воспроизведения.

Так как самостоятельная работа тесно связана с учебным процессом, ее необходимо рассматривать в двух аспектах:

1. аудиторная самостоятельная работа – практические занятия;
2. внеаудиторная самостоятельная работа – подготовка к практическим занятиям (в т.ч. подготовка к практико-ориентированным заданиям и др.).

Основные формы организации самостоятельной работы студентов определяются следующими параметрами:

- содержание учебной дисциплины;
- уровень образования и степень подготовленности студентов;
- необходимость упорядочения нагрузки студентов при самостоятельной работе.

Таким образом, самостоятельная работа студентов является важнейшей составной частью процесса обучения.

Методические указания по организации самостоятельной работы и задания для обучающихся по дисциплине *«Средства коммуникации в учебной и профессиональной деятельности»* обращают внимание студента на главное, существенное в изучаемой дисциплине, помогают выработать умение анализировать явления и факты, связывать теоретические положения с практикой, а также облегчают подготовку к сдаче *зачета*.

Настоящие методические указания позволят студентам самостоятельно овладеть фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю подготовки, опытом творческой и исследовательской деятельности, и направлены на формирование компетенций, предусмотренных учебным планом поданному профилю.

Видами самостоятельной работы обучающихся по дисциплине *«Средства коммуникации в учебной и профессиональной деятельности»* являются:

- самостоятельное изучение тем курса (в т.ч. рассмотрение основных категорий дисциплины, работа с литературой);

- подготовка к практическим (семинарским) занятиям (в т.ч. подготовка к выполнению практико-ориентированных заданий, подготовка реферата);
- подготовка к зачету.

В методических указаниях представлены материалы для самостоятельной работы и рекомендации по организации отдельных её видов.

ОСНОВНЫЕ КАТЕГОРИИ ДИСЦИПЛИНЫ

Тема 1. Сущность коммуникации в разных социальных сферах. Основные функции и виды коммуникации

Коммуникации
Межличностное общение
Речевые способности
Профессиональное общение

Тема 2. Специфика вербальной и невербальной коммуникации

Вербальная коммуникация
Невербальная коммуникация

Тема 3. Эффективное общение

Эффективное общение
Обратная связь
Стиль слушания

Тема 4. Основные коммуникативные барьеры и пути их преодоления в межличностном общении. Стили поведения в конфликтной ситуации

Конфликт
Барьер речи

Тема 5. Виды и формы взаимодействия студентов в условиях образовательной организации

Группа
Коллектив
Групповое давление
Феномен группомыслия
Феномен подчинения авторитету
Обособление
Диктат
Подчинение
Вызов
Выгода
Соперничество
Сотрудничество
Взаимодействие
Взаимопонимание

Тема 6. Формы, методы, технологии самопрезентации

Самопрезентация
Публичное выступление

САМООРГАНИЗАЦИЯ РАБОТЫ С ЛИТЕРАТУРОЙ

Самостоятельное изучение тем курса осуществляется на основе списка рекомендуемой литературы к дисциплине. При работе с книгой необходимо научиться правильно ее читать, вести записи. Самостоятельная работа с учебными и научными изданиями профессиональной и общекультурной тематики – это важнейшее условие формирования научного способа познания.

Основные приемы работы с литературой можно свести к следующим:

- составить перечень книг, с которыми следует познакомиться;
- перечень должен быть систематизированным;
- обязательно выписывать все выходные данные по каждой книге (при написании курсовых и выпускных квалификационных работ это позволит экономить время);

- определить, какие книги (или какие главы книг) следует прочитать более внимательно, а какие – просто просмотреть;

- при составлении перечней литературы следует посоветоваться с преподавателями, которые помогут сориентироваться, на что стоит обратить большее внимание, а на что вообще не стоит тратить время;

- все прочитанные монографии, учебники и научные статьи следует конспектировать, но это не означает, что надо конспектировать «все подряд»: можно выписывать кратко основные идеи автора и иногда приводить наиболее яркие и показательные цитаты (с указанием страниц);

- если книга – собственная, то допускается делать на полях книги краткие пометки или же в конце книги, на пустых страницах просто сделать свой «предметный указатель», где отмечаются наиболее интересные мысли и обязательно указываются страницы в тексте автора;

- следует выработать способность «воспринимать» сложные тексты; для этого лучший прием – научиться «читать медленно», когда понятно каждое прочитанное слово (а если слово незнакомое, то либо с помощью словаря, либо с помощью преподавателя обязательно его узнать). Таким образом, чтение текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации.

От того, насколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия. Грамотная работа с книгой, особенно если речь идет о научной литературе, предполагает соблюдение ряда правил, для овладения которыми необходимо настойчиво учиться. Это серьезный, кропотливый труд. Прежде всего, при такой работе невозможен формальный, поверхностный подход. Не механическое заучивание, не простое накопление цитат, выдержек, а сознательное усвоение прочитанного, осмысление его, стремление дойти до сути – вот главное правило. Другое правило – соблюдение при работе над книгой определенной последовательности. Вначале следует ознакомиться с оглавлением, содержанием предисловия или введения. Это дает общую ориентировку, представление о структуре и вопросах, которые рассматриваются в книге.

Следующий этап – чтение. Первый раз целесообразно прочитать книгу с начала до конца, чтобы получить о ней цельное представление. При повторном чтении происходит постепенное глубокое осмысление каждой главы, критического материала и позитивного изложения; выделение основных идей, системы аргументов, наиболее ярких примеров и т.д. Непременным правилом чтения должно быть выяснение незнакомых слов, терминов, выражений, неизвестных имен, названий. Студентам с этой целью рекомендуется заводить специальные тетради или блокноты. Важная роль в связи с этим принадлежит библиографической подготовке студентов. Она включает в себя умение активно, быстро пользоваться научным аппаратом книги, справочными изданиями, каталогами, умение вести поиск необходимой информации, обрабатывать и систематизировать ее.

Выделяют четыре основные установки в чтении текста:

- информационно-поисковая (задача – найти, выделить искомую информацию);
- усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить, как сами сведения, излагаемые автором, так и всю логику его рассуждений);
- аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему);
- творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

С наличием различных установок обращения к тексту связано существование и нескольких видов чтения:

- библиографическое – просматривание карточек каталога, рекомендательных списков, сводных списков журналов и статей за год и т.п.;
- просмотрное – используется для поиска материалов, содержащих нужную информацию, обычно к нему прибегают сразу после работы со списками литературы и каталогами, в результате такого просмотра читатель устанавливает, какие из источников будут использованы в дальнейшей работе;
- ознакомительное – подразумевает сплошное, достаточно подробное прочтение отобранных статей, глав, отдельных страниц; цель – познакомиться с характером информации, узнать, какие вопросы вынесены автором на рассмотрение, провести сортировку материала;
- изучающее – предполагает доскональное освоение материала; в ходе такого чтения проявляется доверие читателя к автору, готовность принять изложенную информацию, реализуется установка на предельно полное понимание материала;
- аналитико-критическое и творческое чтение – два вида чтения близкие между собой тем, что участвуют в решении исследовательских задач.

Первый из них предполагает направленный критический анализ, как самой информации, так и способов ее получения и подачи автором; второе –

поиск тех суждений, фактов, по которым, или, в связи с которыми, читатель считает нужным высказать собственные мысли.

Из всех рассмотренных видов чтения основным для студентов является изучающее – именно оно позволяет в работе с учебной и научной литературой накапливать знания в различных областях. Вот почему именно этот вид чтения в рамках образовательной деятельности должен быть освоен в первую очередь. Кроме того, при овладении данным видом чтения формируются основные приемы, повышающие эффективность работы с текстом. Научная методика работы с литературой предусматривает также ведение записи прочитанного. Это позволяет привести в систему знания, полученные при чтении, сосредоточить внимание на главных положениях, зафиксировать, закрепить их в памяти, а при необходимости вновь обратиться к ним.

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения.

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала.

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала.

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора.

Конспектирование – краткое и последовательное изложение содержания прочитанного. Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Как правильно составлять конспект? Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта. Выделите главное, составьте план, представляющий собой перечень заголовков, подзаголовков, вопросов, последовательно раскрываемых затем в конспекте. Это первый элемент конспекта. Вторым элементом конспекта являются тезисы. Тезис - это кратко сформулированное положение. Для лучшего усвоения и запоминания материала следует записывать тезисы своими словами. Тезисы, выдвигаемые в конспекте, нужно доказывать. Поэтому третий элемент конспекта - основные доводы, доказывающие истинность рассматриваемого тезиса. В конспекте могут быть положения и примеры. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть

логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Конспектирование –наиболее сложный этап работы. Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы. Конспект ускоряет повторение материала, экономит время при повторном, после определенного перерыва, обращении к уже знакомой работе. Учитывая индивидуальные особенности каждого студента, можно дать лишь некоторые, наиболее оправдавшие себя общие правила, с которыми преподаватель и обязан познакомить студентов:

1. Главное в конспекте не объем, а содержание. В нем должны быть отражены основные принципиальные положения источника, то новое, что внес его автор, основные методологические положения работы. Умение излагать мысли автора сжато, кратко и собственными словами приходит с опытом и знаниями. Но их накоплению помогает соблюдение одного важного правила – не торопиться записывать при первом же чтении, вносить в конспект лишь то, что стало ясным.

2. Форма ведения конспекта может быть самой разнообразной, она может изменяться, совершенствоваться. Но начинаться конспект всегда должен с указания полного наименования работы, фамилии автора, года и места издания; цитаты берутся в кавычки с обязательной ссылкой на страницу книги.

3. Конспект не должен быть «слепым», безликим, состоящим из сплошного текста. Особо важные места, яркие примеры выделяются цветным подчеркиванием, взятием в рамочку, оттенением, пометками на полях специальными знаками, чтобы можно было быстро найти нужное положение. Дополнительные материалы из других источников можно давать на полях, где записываются свои суждения, мысли, появившиеся уже после составления конспекта.

ПОДГОТОВКА К ПРАКТИКО-ОРИЕНТИРОВАННЫМ ЗАДАНИЯМ

Практико-ориентированные задания выступают средством формирования у студентов системы интегрированных умений и навыков, необходимых для освоения профессиональных компетенций. Это могут быть ситуации, требующие применения умений и навыков, специфичных для соответствующего профиля обучения (знания содержания предмета), ситуации, требующие организации деятельности, выбора её оптимальной структуры личностно-ориентированных ситуаций (нахождение нестандартного способа решения).

Кроме этого, они выступают средством формирования у студентов умений определять, разрабатывать и применять оптимальные методы решения профессиональных задач. Они строятся на основе ситуаций, возникающих на различных уровнях осуществления практики и формулируются в виде производственных поручений (заданий).

Под практико-ориентированными заданиями понимают задачи из окружающей действительности, связанные с формированием практических навыков, необходимых в повседневной жизни, в том числе с использованием элементов производственных процессов.

Цель практико-ориентированных заданий – приобретение умений и навыков практической деятельности по изучаемой дисциплине.

Задачи практико-ориентированных заданий:

- закрепление, углубление, расширение и детализация знаний студентов при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- обучение приемам решения практических задач;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Важными отличительными особенностями практико-ориентированных задания от стандартных задач (предметных, межпредметных, прикладных) являются:

- значимость (познавательная, профессиональная, общекультурная, социальная) получаемого результата, что обеспечивает познавательную мотивацию обучающегося;
- условие задания сформулировано как сюжет, ситуация или проблема, для разрешения которой необходимо использовать знания из разных разделов основного предмета, из другого предмета или из жизни, на которые нет явного указания в тексте задания;

- информация и данные в задании могут быть представлены в различной форме (рисунок, таблица, схема, диаграмма, график и т.д.), что потребует распознавания объектов;

- указание (явное или неявное) области применения результата, полученного при решении задания.

Кроме выделенных четырех характеристик, практико-ориентированные задания имеют следующие:

1. по структуре эти задания – нестандартные, т.е. в структуре задания не все его компоненты полностью определены;

2. наличие избыточных, недостающих или противоречивых данных в условии задания, что приводит к объемной формулировке условия;

3. наличие нескольких способов решения (различная степень рациональности), причем данные способы могут быть неизвестны учащимся, и их потребуется сконструировать.

При выполнении практико-ориентированных заданий следует руководствоваться следующими общими рекомендациями:

- для выполнения практико-ориентированного задания необходимо внимательно прочитать задание, повторить лекционный материал по соответствующей теме, изучить рекомендуемую литературу, в т.ч. дополнительную;

- выполнение практико-ориентированного задания включает постановку задачи, выбор способа решения задания, разработку алгоритма практических действий, программы, рекомендаций, сценария и т. п.;

- если практико-ориентированное задание выдается по вариантам, то получить номер варианта исходных данных у преподавателя; если нет вариантов, то нужно подобрать исходные данные самостоятельно, используя различные источники информации;

- для выполнения практико-ориентированного задания может использоваться метод малых групп. Работа в малых группах предполагает решение определенных образовательных задач в рамках небольших групп с последующим обсуждением полученных результатов. Этот метод развивает навыки сотрудничества, достижения компромиссного решения, аналитические способности.

ПРАКТИКО-ОРИЕНТИРОВАННЫЕ ЗАДАНИЯ

1. Организуйте коллективную сетевую деятельность.

Методические указания:

Под организацией **коллективной сетевой деятельности** понимают совместные действия нескольких пользователей в сети электронных коммуникаций, направленные на получение информации. Участники совместной сетевой деятельности могут быть объединены общими целями, интересами, что позволяет им обмениваться мнениями, суждениями, а также совершать действия с различными объектами, такими как фотографии, программы, записи, статьи, представленными в цифровом виде.

Подобное взаимодействие может заключаться в различных его видах, таких как:

- - общение;
- - обмен данными;
- - организация трудовой деятельности;
- - совместное времяпрепровождение за сетевыми развлечениями.

Рассмотрим каждый из них. Одним из примеров организации **общения** в сети Интернет могут служить популярные на сегодняшний день сообщества **Livejournal** (www.livejournal.ru), **Facebook** (www.facebook.com), **Twitter** (<http://twitter.com>) и др.

По своей сути это социальные сети, которые работают в режиме реального времени, позволяя участникам взаимодействовать друг с другом. Так, социальная сеть Livejournal (Живой журнал) предоставляет возможность публиковать свои и комментировать чужие записи, вести коллективные блоги («сообщества»), получать оперативную информацию, хранить фотографии и видеоролики, добавлять в друзья других пользователей и следить за их записями в «ленте друзей» и др.

Facebook позволяет создать профиль с фотографией и информацией о себе, приглашать друзей, обмениваться с ними сообщениями, изменять свой статус, оставлять сообщения на своей и чужой «стенах», загружать фотографии и видеозаписи, создавать группы (сообщества по интересам).

Система Twitter позволяет пользователям отправлять короткие текстовые заметки, используя web-интерфейс, sms-сообщения, средства мгновенного обмена сообщениями (например, Windows Live Messenger), сторонние программы-клиенты. Отличительной особенностью Твиттера является публичная доступность размещенных сообщений, что роднит его с **блогами** (онлайн-дневник, содержимое которого, представляет собой регулярно обновляемые записи — **посты**).

Другим способом общения, безусловно, является **электронная почта**. Принципы создания ящика электронной почты подробно рассматривались в практикуме параграфа 2.12. При всех своих плюсах электронная почта не позволяет организовать двусторонний оперативный диалог, максимально приближенный к обычному разговору. Отправив письмо, человек уверен, что оно оперативно будет доставлено в ящик адресата, но будет ли получен быстрый ответ? Кроме того, переписка может растянуться, что сводит к минимуму решение возможных актуальных проблем человека в настоящий момент времени.

Именно поэтому возникла необходимость в самостоятельном классе программ, которые выполняли бы две основные задачи:

1. Показать, находится ли собеседник в данный момент в сети Интернет, готов ли он общаться.
2. Отправить собеседнику короткое сообщение и тут же получить от него ответ.

Такие программы получили название IMS (англ. Instant Messengers Service — служба мгновенных сообщений). Часто такие программы называют **интернет-пейджерами**. В качестве примера подобных программ можно привести Windows Live Messenger, Yahoo!Messenger, ICQ.

Так, программа Windows Live Messenger является одним из компонентов Windows Live — набора сетевых служб от компании Microsoft. Ранее мы познакомились с такими его модулями, как Семейная безопасность и Киностудия. Доступ к Messenger можно получить по адресу <http://download.ru.msn.com/wl/messenger>, либо через кнопку **Пуск** на своем персональном компьютере (предварительно установив основные компоненты службы Windows Live).

В настоящее время произошла интеграция Messenger и программы Skype, функции которой будут рассмотрены позже.

Чтобы начать «разговор», достаточно выполнить двойной щелчок мыши на имени собеседника и ввести сообщение в соответствующее окно. Если друга нет на месте, можно оставить ему сообщение, и он увидит его, когда снова войдет в программу.

Коммуникацию в реальном масштабе времени возможно осуществить с помощью **чатов** (англ. Chatter — болтать). Если ваш компьютер оснащен видеокамерой, вы сможете начать видеочат. Одной из наиболее интересных особенностей видео-чата в Messenger является то, что он позволяет делать через Интернет все, что ранее можно было делать только при личном общении. Например, можно легко обмениваться фотографиями и видеть, как собеседник реагирует на них.

Теперь рассмотрим, каким образом можно организовать коллективную сетевую деятельность, связанную с **обменом данными**. Сразу отметим, что для передачи или открытия доступа к файлам в локальной сети используются стандартные возможности операционной системы компьютера. Для этого достаточно в настройках определенной директории открыть общий доступ на чтение или запись другими пользователями сети.

В настоящее время популярнейшим способом обмена данными является размещение файлов на различных видеохостингах и в социальных сетях. **Хостинг** — это услуга по предоставлению вычислительных мощностей для размещения информации на сервере, постоянно находящемся в сети Интернет. Для размещения видеофайлов, как правило, используются такие крупные видеохостинги, как YouTube (www.youtube.com), Rutube (<http://mtube.ru>). Социальные сети, например Одноклассники (www.odnoklassniki.ru), ВКонтакте (<http://vk.com>) и др., также можно использовать для размещения видеоматериалов.

Хранение, обмен файлов возможно организовать и с помощью облачных сервисов, таких как Яндекс.Диск, SkyDrive, iCloud и т.д. Перечислим ряд достоинств подобного способа организации работы:

- не требуется денежных вложений - сервисы бесплатны;
- возможность резервного хранения данных;
- доступность информации из любой точки мира с разных устройств, подключенных к Интернету;
- пользователь самостоятельно определяет доступность к файлам другим людям;
- большой размер облачного хранилища (7-10 Гб);
- информация не привязана к одному компьютеру;
- доступ к файлам, хранящимся на устройствах с разными аппаратными платформами (Windows, Android, iOS).

В качестве примера рассмотрим работу с программой Яндекс.Диск, которую предварительно следует установить на свой компьютер с адреса <http://disk.yandex.ru/download>. После инсталляции программы на вашем устройстве создается папка Яндекс.Диск, в которой будет находиться ряд папок, таких как Документы, Музыка, Корзина. Теперь, после того как мы добавим, изменим или удалим файл в папке Яндекс.Диск на своем компьютере, то же самое автоматически произойдет на серверах Яндекс, т. е. происходит процесс синхронизации.

Поделиться файлом с друзьями через web-интерфейс можно, выполнив следующие действия:

1. Зайти в свой почтовый ящик на сервисе Яндекс.

2. Выполнив команду **Файлы/Документы**, выделить нужный файл из списка.

3. Установить переключатель на панели предпросмотра в положение **Публичный** и нажать на одну из кнопок, расположенных ниже, что гарантирует публикацию ссылки на файл в одной из социальных сетей (ВКонтакте, Facebook и т.д.) либо отправку по электронной почте (рис. 1).



Рис. 1. Ссылка на файл

Другой возможностью публикации ссылки на файл - получение ее через ОС Windows. В этом случае порядок действий следующий:

1. Открыть папку Яндекс.Диск.
2. Выполнить щелчок правой кнопкой мыши на нужном файле.
3. В контекстном меню выбрать пункт **Яндекс.Диск: Скопировать публичную ссылку**.

Теперь в буфере обмена находится ссылка на файл, например, <http://yadi.Sk/d/91nV8FjiOYnX>, с которой вы можете поделиться со своими друзьями.

Перейдем к описанию организации **трудовой деятельности** как способа совместного сетевого взаимодействия. Она может выглядеть самой разной, от простого общения в видеоконференциях, заканчивая использованием серьезных корпоративных решений для управления рабочим процессом в компании. Примерами таких решений являются:

1. 1С-Битрикс: Корпоративный Портал (<http://www.lc-bitrix.ru/products/intranet/>) — система управления внутренним информационным ресурсом компании для коллективной работы над задачами, проектами и документами.
2. Мегатлан (www.megaplan.ru) — онлайн-сервис для управления бизнесом.
3. TeamLab (www.teamlab.com/ru) — многофункциональный онлайн-сервис для совместной работы, управления документами и проектами.
4. BaseCamp (<http://basecamp.com>) — онлайн-инструмент для управления проектами, совместной работы и постановки задач по проектам.

Рассмотрим эти решения на примере облачного сервиса **Мегатлан**, который относится к модели **SaaS** (англ. Software as a service — программное обеспечение как услуга). В рамках модели SaaS заказчики платят не за владение программным обеспечением как таковым, а за его аренду (т. е. за его использование через web-интерфейс). Таким образом, в отличие от классической схемы лицензирования программного обеспечения заказчик несет сравнительно небольшие периодические затраты (от 150 до 400 руб./мес.), и ему не требуется инвестировать значительные средства в приобретение ПО и аппаратной платформы для его развертывания, а затем поддерживать его работоспособность.

Используя на предприятии Мегатлан, можно получить множество современных эффективных средств управления персоналом компании, в частности:

- выстроить иерархическую структуру предприятия, прояснить уровни подчинения, сделать связи сотрудников внутри предприятия логичными и понятными каждому;
- система управления персоналом на предприятии позволит каждому руководителю контролировать деятельность своих подчиненных в режиме реального времени. Кроме того,

можно получать актуальную информацию, даже не находясь в офисе — для этого достаточно иметь доступ в Интернет;

- получить возможность обмениваться документами, выкладывать в общий доступ бизнес-планы, презентации, проекты и распоряжения, ускоряя обмен информацией внутри предприятия;

- системы обмена сообщениями и корпоративный форум делают общение, как деловое, так и личное, более живым и эффективным. Кроме того, выходящая по ходу исполнения задачи, зафиксированные в Мегаплане, позволяют анализировать ход работы над проектом.

Зарегистрировавшись на вышеуказанном сайте, вы получите бесплатный доступ для знакомства с сервисом Мегаплан. Из трех решений предлагаемых компанией, а именно Совместная работа, Учет клиентов и Бизнес-менеджер, выберите первое — **Совместная работа**. Такой выбор дает возможность эффективно управлять проектами, задачами и людьми. Выбрав модуль **Сотрудники**, добавьте несколько сотрудников, заполнив их личные карточки. Много информации в карточки заносить необязательно, их всегда можно отредактировать, при этом не забывая нажимать на кнопку **Сохранить**. Заполненный модуль **Сотрудники** представлен на рис. 2.



Рис. 2. Модуль Сотрудники

Заполнив базу сотрудников, отметив все необходимые сведения в картотеке, вы получаете автоматизированную систему управления персоналом компании, которая более оперативно, чем любой менеджер по кадрам, будет оповещать вас обо всех изменениях, напоминать о днях рождения, давать доступ к картотеке и персональным сообщениям.

Теперь создайте отделы своей виртуальной организации. Для этого, находясь в модуле **Сотрудники**, выберите блок **Структура**, а в нем ссылку **Добавить отдел**. Чтобы добавить сотрудника в отдел, его надо перетащить мышью из списка **Нераспределенные**. После этого следует установить связь «Начальник-Подчиненный», используя ссылки **Начальники**, **Подчиненные**. Подобная ситуация представлена на рис. 3.

Красные стрелки на схеме обозначают вашу подчиненность, а зеленые — сотрудники подчиняются вам.

Для того чтобы организовать взаимодействие в команде, выберите модуль **Задачи** и поставьте перед каждым сотрудником задачу, указав сроки ее выполнения. Сотрудник может принять или отклонить задачу, делегировать ее своему подчиненному, комментировать задачу, оперировать списком своих задач (распечатывать, сортировать по признакам). Он может даже провалить задачу — и это немедленно станет известно всем, кто с ней связан.

Используя модуль **Документы**, попробуйте создать несколько текстовых документов (их объем не может превышать 300 Мб). Также имеется возможность импортировать имеющиеся документы, которые Мегаплан будет сортировать по типам: текстовые

документы, презентации, PDF-файлы, таблицы, изображения и др. Таким образом, можно хранить общие для всей компании договоры, банки, анкеты и другие важные файлы.



Рис. 3 Организационная структура предприятия

Модуль **Обсуждение** представляет собой корпоративный форум, в рамках которого можно рассматривать любые вопросы. Обсуждение тем может происходить в нескольких уже созданных разделах, а именно Новости, Отдых, Работа. Подобная ситуация представлена на рис. 4.

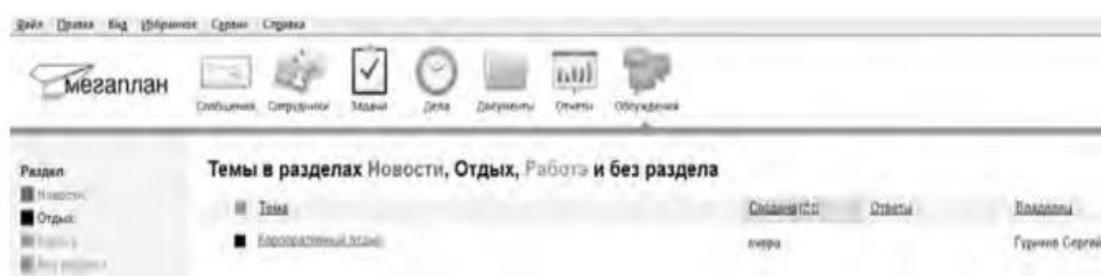


Рис. 4.Создание темы в модуле Обсуждение

Создайте несколько тем, воспользовавшись кнопкой **Добавить**. Обратите внимание на то, что вы можете ограничить просмотр обсуждаемых тем отдельным сотрудникам и группам. Корпоративный форум делает общение внутри компании более открытым. Возможность общения онлайн между сотрудниками, встреча которых могла бы и не произойти в реальной жизни, развивает неформальные отношения, вследствие которых совместная работа над проектами становится более комфортной. Работа над проектом, созданным в виртуальной среде, существенно упрощается за счет системы обмена сообщениями (модуль **Сообщения**), совместной работы, обработки файлов, находящихся в общем доступе.

Итак, освоение базовых функциональных операций в процессе работы с Мегалланом происходит очень быстро. С учетом того, что бесплатная версия продукта позволяет зарегистрировать трех пользователей, можно организовать сетевое взаимодействие, создав учебное предприятие и тем самым, усовершенствовать навыки взаимодействия исполнителей и руководителей в рабочем процессе.

Совместное времяпрепровождение за сетевыми развлечениями — последний вид сетевого взаимодействия, рассматриваемого нами. Сетевыми развлечениями в основном являются компьютерные игры. Вид взаимодействия в играх может быть различным: игроки могут соперничать друг с другом, могут быть в команде, а в некоторых играх возможны оба

вида взаимодействия. Соперничество может выражаться как напрямую, например игра в шахматы, так и в таблице рейтингов в какой-нибудь браузерной игре.

Существует особый жанр игр MMORPG (англ. Massive Multiplayer Online Role-playing Game, массовая многопользовательская онлайн-ролевая игра) — разновидность онлайн-ролевых игр, позволяющая тысячам людей одновременно играть в изменяющемся виртуальном мире через Интернет. Сообщество любителей игр в жанре MMORPG зарегистрировано в сети Интернет по адресу www.mmorpg.su.

Подобные игры, как правило, построены на технологии «клиент-сервер», но есть разновидности, где в качестве клиента выступает обычный браузер. Игрок в такой игре представляется своим **аватаром** — виртуальным представлением его игрового персонажа. Создатели игры поддерживают существование игрового мира, в котором происходит действие игры и который населен ее персонажами.

Когда геймеры попадают в игровой мир, они могут в нем выполнять различные действия вместе с другими игроками со всего мира. Разработчики MMORPG поддерживают и постоянно развивают свои миры, добавляя новые возможности и доступные действия для того, чтобы «гарантировать» интерес игроков. Яркими представителями подобного рода игр на сегодняшний день являются EverQuest, World of Warcraft, Anarchy Online, Asheron's Call, Everquest II, Guild Wars, Ragnarok Online, Silkroad Online, The Matrix Online, City of Heroes.

Задания:

а)Создайте свой аккаунт (если вы его не имеете) в одной из социальных сетей, например Livejournal или Facebook. Выполните скриншоты своего блога. Результат отправьте на электронную почту преподавателя.

б)Используя программу Windows Live Messenger, добавьте в друзья (по предварительной договоренности) своего преподавателя и свяжитесь с ним в режиме реального времени либо оставьте ему сообщение.

в)Установите на свой компьютер программу Яндекс.Диск. Предоставьте доступ к нескольким файлам своему преподавателю.

г)Создайте учебное предприятие, используя облачный сервис Мегатлан. Заполните информацией все имеющиеся в программе модули. Установите связи между отделами. Пригласите нескольких своих друзей в проект. Продемонстрируйте результат преподавателю, открыв ему доступ.

д)Напишите краткий отчет о результатах своей работы по созданию виртуального предприятия, указав в нем этапы его создания, результаты совместной сетевой деятельности.

е)Являетесь ли вы участником какой-либо игры в жанре MMORPG? Если да, расскажите об основных правилах той игры, в которой вы участвуете. Каким образом происходит ваше взаимодействие в ней с друзьями?

3. Организация форумов

Методические указания

В настоящее время перед каждым образовательным учреждением стоит задача формирования открытой информационной образовательной среды. Эффективным механизмом является использование коммуникационных возможностей сети Интернет. В частности, организация на сайтах или в информационных системах образовательных учреждений форумов (дискуссий).

Форум — это web-страница, созданная на основе клиент-серверной технологии для организации общения пользователей сети Интернет. Концепция форума основана на создании разделов, внутри которых происходит обсуждение различных тем в форме сообщений. От чата форум отличается тем, что общение может происходить не в реальном времени. Таким образом, человек имеет возможность подумать над своим ответом или над создаваемой темой.

По методу формирования набора тем форумы бывают:

- **тематические.** В рамках таких форумов пользователи обсуждают предварительно опубликованную статью, новость СМИ и т.д. Обсуждение происходит в одной или нескольких темах;

- **проблемные.** Для обсуждения предлагается ряд проблемных вопросов (тем). Обсуждение каждой проблемы происходит в своей ветке. Чаще всего в подобных типах форумов пользователь не имеет права создавать новую тему;

- **постоянно действующие форумы.** Форумы поддержки (помощи). По такому принципу строятся форумы технической поддержки, различные консультации и пр. Чаще всего это форумы с динамическим списком тем, где простые участники могут создавать новую тему в рамках тематики форума.

Форумы функционируют согласно определенным правилам, которые определяют администраторы и модераторы. **Администратор форума** следит за порядком во всех разделах, контролирует общение на ресурсе и соблюдение правил сайта. **Модератор форума** чаще всего следит за порядком в конкретном разделе, имеет более узкие права, чем администратор. Его основная задача — увеличивать популярность форума, количество участников и число интересных обсуждений. Дополнительные задачи:

- стимулировать появление новых интересных тем;
- стимулировать общение на форуме;
- не допускать конфликтных ситуаций на форуме, а в случае их возникновения — уметь найти выход из сложной ситуации;
- при появлении в темах **спама** (рассылка коммерческой и иной рекламы или иных видов сообщений (информации) лицам, не выразившим желания их получать) немедленно сообщать об этом администратору сайта;
- следить за культурой сетевого общения.

Для каждого конкретного форума администратором могут быть созданы свои правила, но в целом их можно свести к следующим:

1. На форумах приветствуется поддержание дискуссии, обмен опытом, предоставление интересной информации, полезных ссылок.

2. Не нужно вести разговор на «вольные» темы и размещать бессодержательные (малосодержательные) или повторяющиеся сообщения. Под бессодержательными (малосодержательными) понимаются, в частности, сообщения, содержащие исключительно или преимущественно эмоции (одобрение, возмущение и т. д.).

3. Желательно проверять грамотность сообщений (например, редактором Microsoft Word) — ошибки затрудняют понимание вопроса или ответа и могут раздражать участников обсуждения.

4. Длинные сообщения желательно разбивать на абзацы пустыми строчками, чтобы их было удобно читать.

5. Запрещается размещать заведомо ложную информацию.

6. Не рекомендуется публиковать сообщения, не соответствующие обсуждаемой теме, в том числе личные разговоры в ветках форума.

7. Не следует писать сообщения сплошными заглавными буквами, так как это эквивалентно повышению тона, а также латинскими буквами. При этом сообщение считается нарушающим данное правило, если такого рода текстом набрано более трети всего сообщения.

8. Участники форума не должны нарушать общепринятые нормы и правила поведения. Исключено употребление грубых слов и ненормативной лексики, выражение расистских, непристойных, оскорбительных или угрожающих высказываний, нарушений законодательства в области авторского права или сохранности конфиденциальной информации.

9. Запрещено публично обсуждать нелегальное использование (в том числе взлом) программного обеспечения, систем безопасности, а также публикацию паролей, серийных номеров и адреса (ссылки), по которым можно найти что-либо из вышеназванного.

- - сообщить по электронной почте друзьям о новом форуме и пригласить их поучаствовать в форуме;
- - поместить ссылки на форум на других сайтах, форумах и в поисковых системах.

Для администрирования вновь созданного форума необходимо ввести имя пользователя (Admin) и пароль, который вы выбрали при создании форума. После этого вы получаете доступ к ссылке **Панель администратора**, расположенной внизу страницы, которая имеет несколько вкладок (рис. 6).



Рис. 6. Вкладки Панели администратора

Вкладка **Главная** отображает информацию по статистике созданных сообщений, количеству пользователей и тем. Здесь же можно воспользоваться практическими советами по повышению посещаемости созданного форума. Попробуйте пригласить на созданный форум своих друзей, знакомых, с помощью ссылки **Адреса Email**, вводя в соответствующее поле их электронные адреса. Максимальное число приглашений, отправляемых за один раз, — десять.

Вкладка **Общие настройки** позволяет сконфигурировать форум в соответствии с личными целями администратора. В частности, можно изменить название сайта, его описание, определить конфигурацию защиты форума, определить E-mail администратора.

С помощью раздела **Категории и форумы** создайте свои форумы, определите порядок их вывода с помощью соответствующих кнопок (**Сдвинуть вверх**, **Сдвинуть вниз**). **Категория** представляет собой совокупность форумов, объединенных общей тематикой. Один из возможных примеров создания форумов приведен на рис. 7.

Сделанные изменения доступны для просмотра после нажатия на кнопку **Просмотр форума**. Находясь на вкладке **Общие настройки**, перейдите в раздел **Раскрутка форума** и выберите пункт **Поисковые системы**. Введите информацию для ваших мета-тегов, чтобы улучшить позицию вашего форума в поисковых системах. **Мета-теги** — это невидимые коды, используемые поисковиками для индексации и позиционирования вашего форума. Зарегистрируйте ваш форум в основных поисковых системах: Yandex, Google, Rambler.



Рис. 7. Структура форумов

Используя вкладку **Оформление**, поэкспериментируйте с различными стилями для того, чтобы повысить привлекательность форума. Здесь же можно поменять версию «движка» форума.

Будучи администратором вашего форума, вы являетесь его единственным полноправным хозяином и полностью контролируете его. С помощью вкладки **Пользователи & Группы** создайте группу модераторов, ответственных за соблюдение установленных вами правил (правил орфографии, правил поведения на форуме и т.д.).

Перейдите на вкладку **Модули**. Здесь вы можете добавить к вашему форуму такие модули, как портал, календарь, галерея, чат или листы персонажей. Выберите ссылку **Портал**. Появится информация о том, что портал не установлен. Нажмите ссылку — установить. Внешний вид созданного портала представлен на рис. 8.



Рис. 8. Созданный портал

На вкладке **Модули** попробуйте поработать с виджетами (гаджетами) форума, из которых и состоит портал. **Виджет** — это элемент интерфейса, предназначенный для облегчения доступа к информации.

Добавьте/удалите стандартные виджеты форума (Поиск, Календарь, Новости, Последние темы, Самые активные пользователи и др.), отслеживая изменения нажатием кнопки **Просмотр портала**. Оставьте наиболее удачный, с вашей точки зрения, вариант.

Итак, мы приобрели первоначальные практические навыки создания собственного форума и выполнили действия, направленные на увеличение его посещаемости. Кроме того, необходимо создать ссылку на форум с главной страницы сайта учебного заведения. Следует отметить, что, для того чтобы созданный форум не оставался в статичном виде, необходима большая работа администратора, модераторов по его поддержанию.

Альтернативным способом организации форумов является их развертывание в информационной системе учебного заведения. На современном отечественном рынке автоматизированных информационных систем управления учебным процессом представлено достаточно большое количество решений. Свой выбор остановим на ИС ModEUS (<http://modeus.krf.ane.ru/index.php>), которая разработана с учетом специфики российского образования и обеспечивает автоматизацию учебного процесса, в том числе и дистантного (учет учебного процесса, его планирование и публикация, подготовка отчетной документации).

После регистрации в системе ModEUS, нужно выбрать ссылку **Дискуссии**. Вы можете организовать дискуссию (форум) по любому из находящихся в системе курсов, щелкнув мышью по его названию.



Рис. 9. Страница Дискуссии в ИС ModEUS

Создадим новую тему, нажав одноименную кнопку. Впишем в соответствующие поля название темы и вопрос, предлагаемый для обсуждения. Подобная ситуация представлена на рис. 437. Кроме того, мы имеем возможность прикрепить текстовый файл объемом не более 16 Мб, например список вопросов к экзамену.

После нажатия на кнопку **Создать** тема дискуссии отображается в системе (рис. 10), и любой из студентов может принять участие в ее обсуждении.

Таким образом, можно определить преимущества создания форума в информационной системе учебного заведения:

- - отсутствует необходимость иметь практические навыки работы по созданию web-страниц;
- - нет необходимости заботиться о раскрутке форума - студенты и преподаватели постоянно работают в системе.

В то же время есть и ряд недостатков, в частности:

- - форум доступен исключительно для студентов и преподавателей учебного заведения, в котором функционирует информационная система;
- - стандартизированный типовой интерфейс для всех выполняемых функций;
- - нет возможности организовать дискуссию на вольную тему.



Рис. 10 Создание новой темы



Рис. 11. Создана тема для дискуссии

Использование тестирующих систем в локальной сети образовательного учреждения

Теперь познакомимся с возможностями ИС ModEUS для **организации тестирования студентов в локальной сети образовательного учреждения**. Использование тестирования как наиболее объективного метода оценки качества образования широко используется в учебных заведениях России. Полнота охвата проверкой требований к уровню подготовки студентов предполагает методику конструирования тестовых заданий закрытого и открытого типа. К тестовым заданиям **закрытого типа** относятся задания, предполагающие выбор верного ответа из предложенных вопросов. Тестовые задания **открытого типа** требуют конструирования ответов с кратким и развернутым ответом. И тот, и другой тип заданий успешно реализуются в ИС ModEUS.

Прежде чем создать тестовое задание, необходимо зайти в один из учебных курсов, находящихся в репозитории (хранилище данных), нажав кнопку **Курсы** в главном меню. Под «курсом» в ИС ModEUS понимается дисциплина, находящаяся в учебном плане.

Найдем в списке **Занятия курса** требуемое занятие и нажмем ссылку **Список заданий**, находящуюся справа от поля **Тип**. Для того чтобы добавить задание в занятие, нажмем кнопку **Добавить**. Подобная ситуация представлена на рис. 11.

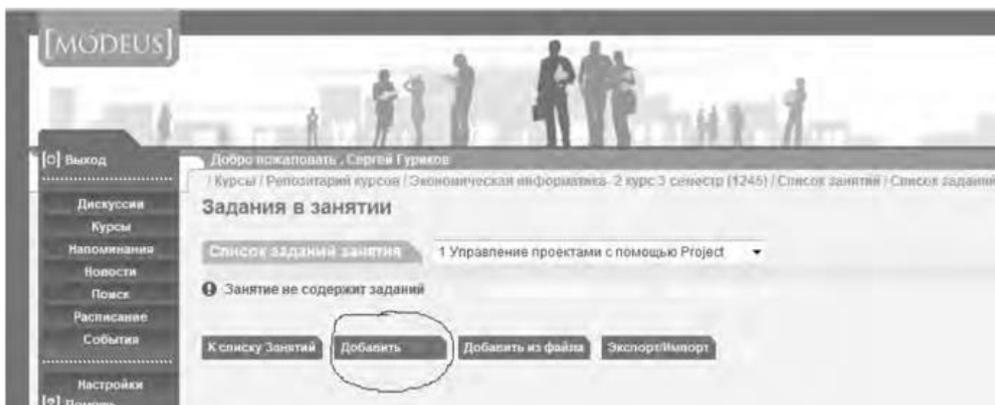


Рис.12. Добавление задания

Тип задания можно выбрать из раскрывающегося списка (рис. 12), кроме того, можно дать название новому заданию, установить балл и выбрать количество попыток сдачи.

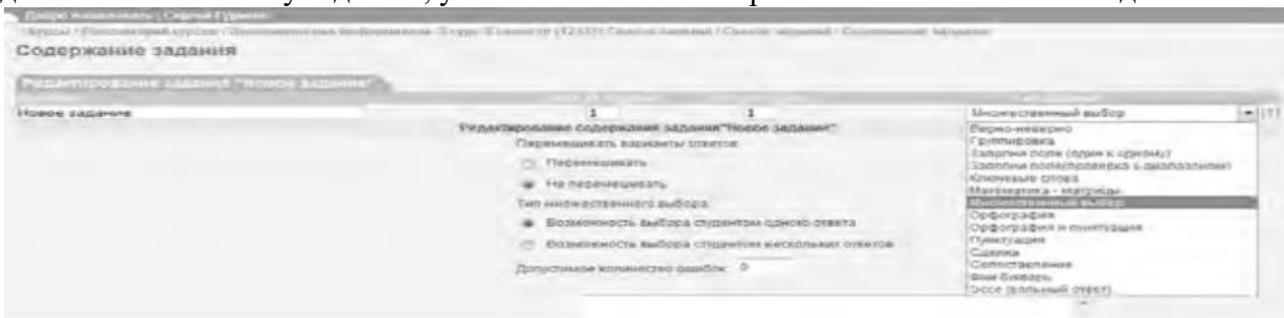


Рис.13. Выбор типа задания

Рассмотрим несколько примеров формирования вопросов закрытого и открытого типа в ИС ModEUS.

Тестовое задание со множественным выбором верных ответов (закрытый тип). Данный тип задания дает вам возможность задать вопрос и варианты ответов на него, из которых обучающийся должен выбрать верный (рис. 14). Правильным может быть один или несколько вариантов. Для того чтобы наполнить задание, выполните следующие действия:

- - в опции **Перемешивать варианты ответов** поставьте метку в поле **Перемешивать**, если вы хотите, чтобы указанные вами варианты ответов выводились на экран в различном порядке, поставьте метку в поле **Не перемешивать**, если варианты ответов должны выводиться всегда в одинаковом порядке;
- - в опции **Тип множественного выбора** поставьте метку в поле **Возможность выбора студентом одного ответа**, если обучающийся из предложенных вариантов ответов может выбрать только один верный, поставьте метку в поле **Возможность выбора студентом нескольких ответов**, если обучающийся может выбрать несколько верных ответов;
 - - введите текст задания в поле **Текст задания**;
 - - в случае если в задании присутствует приложение, укажите путь к этому приложению, нажав на кнопку **Обзор...** и указав путь к файлу на жестком или сетевом диске. Приложением может быть документ любого формата, например изображение;
 - - введите тексты вариантов ответов в соответствующие поля;
 - - для добавления нового поля под вариант ответа нажмите на кнопку



- каждый вариант ответа может быть дополнен приложением. Для добавления к варианту ответа приложения укажите путь к нему в поле **Добавить приложение**, нажав на



кнопку и указав путь к файлу на жестком

или сетевом диске;

- установите флажки напротив одного или нескольких правильных вариантов ответа;

- нажмите на кнопку **Зафиксировать** для сохранения задания в базе данных;

- нажмите на кнопку **Сохранить и добавить новое**; чтобы сохранить задание и сразу

перейти к составлению нового задания.

Название	Балл за задание	Попыток сдачи	Тип задания
Задание 6	1	1	Множественный выбор

Редактирование содержания задания "Задание 6"

Перемешивать варианты ответов:

Перемешивать

Не перемешивать

Тип множественного выбора:

Возможность выбора студентом одного ответа

Возможность выбора студентом нескольких ответов

Текст задания:

На каком уровне семиуровневой модели ISO происходит передача кадра данных между узлами. В качестве адресов используются MAC-адреса

Добавить приложение: Обзор...  

Варианты ответов:

1 физический уровень

Добавить приложение: Обзор...  

2 канальный уровень

Добавить приложение: Обзор...

3 сетевой уровень

Добавить приложение: Обзор...

4 транспортный уровень

Добавить приложение: Обзор...

5 сеансовый уровень

Добавить приложение: Обзор...

6 уровень представления

Добавить приложение: Обзор...

7 прикладной уровень

Добавить приложение: Обзор...

Добавить ответ

Рис. 14. Создание задания со множественным выбором верных ответов

Тестовое задание с добавлением слова (открытый тип). Данный тип задания (рис. 15) дает вам возможность задать вопрос, на который обучающийся должен ответить, введя ответ с клавиатуры в виде текста, цифры, слова, математической формулы и т.д. Для того чтобы наполнить задание, выполните следующие действия:

- - введите текст задания в поле **Текст задания**;
- - текст задания может представлять собой текст или текст в сочетании с

приложением. Чтобы добавить приложение (изображение или документ), нажмите на кнопку **Обзор...**; находящуюся под полем **Текст задания**, и укажите путь к файлу на жестком или сетевом диске;

- - в поле **Вопрос** введите вопрос, на который должен ответить обучающийся;
- - в поле **Ответ** укажите правильный ответ;

- в пределах одного задания вы можете задать обучающемуся несколько вопросов. Для

добавления вопроса нажмите на кнопку **Добавить вопрос**;

- нажмите на кнопку **Зафиксировать** для сохранения задания в базе данных;

- нажмите на кнопку **Сохранить и добавить новое**, чтобы сохранить задание и сразу перейти к составлению нового задания.

Рис. 15 Создание задания с добавлением слова

Кроме рассмотренных типов заданий, в ИС ModEUS существует и ряд других, в частности:

Верно - неверно. Данный тип задания предоставляет возможность обучающемуся выбрать один из вариантов ответа («верно» или «неверно») на поставленный вопрос.

Группировка. В данном типе задания обучающемуся необходимо распределить заданный список понятий по группам.

Заполни поле (проверка с диапазоном). Данный тип задания дает возможность задать вопрос, на который обучающийся должен ответить, введя с клавиатуры числовой ответ.

Сопоставление. Проверяется способность обучающихся сопоставить понятия по указанному принципу.

Эссе. Обучающийся отвечает в свободной форме на поставленный преподавателем вопрос. Вопрос может быть представлен в виде текста или любого другого документа.

Следует отметить, что в ИС ModEUS можно задать количество вопросов, время на проведение тестовых заданий, а также **мощность теста**. Мощность определяет количество заданий, которые будут предложены студенту для выполнения. Например, если в группе заданий десять вариантов заданий, а мощность группы равна пяти, то студенту будут предложены для выполнения пять заданий из десяти. После проведения тестирования в

информационной системе происходит автоматическое формирование оценок на основании выполненных студентами заданий.

Итак, мы завершили рассмотрение возможностей информационной системы, работающей в локальной сети учебного заведения для организации форумов и проведения тестирования студентов.

Настройка видео web-сессий

В настоящее время миллионы пользователей во всем мире используют видеосвязь с помощью сети Интернет для общения друг с другом. Достоинства такого способа общения очевидны: есть возможность слышать и визуально наблюдать собеседника, находящегося, возможно, за тысячи километров. Для обеспечения полноценной видеосвязи для захвата и воспроизведения видео и звука могут использоваться как встроенные в компьютер камера, микрофон или динамик, так и внешние устройства, такие как web-камера, головная гарнитура, а также следует обеспечить высокоскоростной доступ к Интернету.

Взаимодействие собеседников при организации видео web-сессий возможно в нескольких направлениях: видеоконференция и видеотелефония.

1. Видеоконференция — это технология интерактивного взаимодействия двух и более человек, при которой между ними происходит обмен информацией в режиме реального времени. Существует нескольких видов видеоконференций:

- **симметричная (групповая)** видеоконференция позволяет проводить сеансы показа презентаций или рабочего стола;
- **асимметричная** видеоконференция используется для дистанционного образования. Позволяет собрать в конференции множество участников таким образом, что все они будут видеть и слышать одного ведущего, он, в свою очередь, всех участников одновременно;
- **селекторное видеосовещание** — рассчитано на взаимодействие большой группы участников, при котором пользователи имеют возможность активно обсуждать действия при чрезвычайных ситуациях, оперативно решать текущие вопросы.

Для эффективной организации проведения web-конференций, маркетинговых презентаций, онлайн-обучения, совещаний и любых других видов онлайн-встреч существует ряд программных решений. В качестве примера можно привести программы Mirapolis Virtual Room (<http://virtualroom.ru/>), ВидеоМост (www.videomost.com), TrueConf Online (<http://trueconf.ru/>) и др.

2. Видеотелефония — реализуется посредством сеанса видеосвязи между двумя пользователями, во время которого они могут видеть и слышать друг друга, обмениваться сообщениями и файлами, вместе работать над документами и при этом находиться в разных местах в комфортной для себя обстановке.

Для того чтобы общаться с близкими и друзьями, можно бесплатно совершать видеозвонки с помощью таких программ, как Skype (<http://www.Skype.com/intl/ru/get-skype>), Mail.ru Агент (<http://agent.mail.ru>) и ряд других.

Для того чтобы проверить наличие встроенной web-камеры на компьютере, достаточно войти в меню **Пуск**, выбрать **Компьютер**, щелкнуть на нем правой кнопкой мыши и в контекстно-зависимом меню нажать пункт **Свойства**. Далее следует выбрать пункт меню **Диспетчер устройств**, а в нем пункт **Устройства обработки изображений**. Наличие в нем устройства, например, USB 2.0 Camera свидетельствует о наличии web-камеры.

Кроме того, в документации к компьютеру (Руководство пользователя) или другому устройству должны быть приведены сведения об установленных в систему устройствах и, в частности, инструкция по использованию встроенной камеры и программному обеспечению, отвечающему за данное устройство.

Одной из таких популярных утилит является ArcSoft WebCam Companion — пакет приложений для взаимодействия с web-камерой, который позволяет захватывать, редактировать изображения и записывать видео. Самостоятельно проведите ее инсталляцию,

воспользовавшись web-адресом <http://arcsoft-webcam-companion.en.softonic.com>. После установки данной программы на компьютер ее можно запустить на выполнение командой **Пуск/Все программы/ArcSoft WebCam Companion/WebCam Companion**. Интерфейс программы представлен несколькими разделами: **Захват**, **Маска**, **Забавная рамка**, **Правка**, **Монитор**, **Другие приложения** (рис. 16).



Рис. 16. Пункты меню программы ArcSoft WebCam Companion

Выберем значок **Захват**, а в нем пункт меню **Параметры web-камеры**. Откроется окно, представленное на рис. 17.

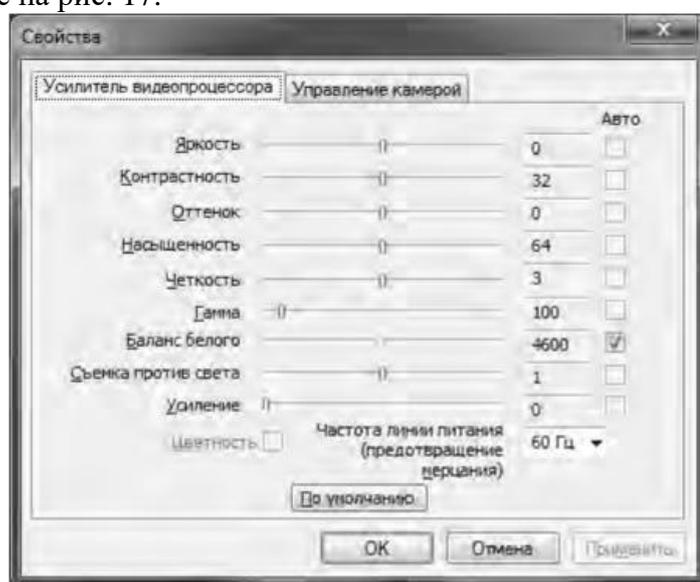


Рис. 17. Окно Свойства web-камеры

Как видно из рис. 17, в данном окне можно изменить основные параметры настройки web-камеры, одновременно наблюдая за результатом на экране. При желании настройки можно вернуть в исходное состояние, нажав на кнопку **По умолчанию**.

Теперь поговорим о том, как организовать web-сессию в такой популярной программе, как Skype. Ее большим преимуществом является такой факт, что звонки между абонентами являются бесплатными. Однако, если вы делаете звонок на мобильный или стационарный телефон, вам потребуется позаботиться о том, чтобы на вашем счете были деньги. Положить деньги на оплату разговоров в Skype вы можете с использованием такого сервиса, как Яндекс.Деньги (<https://money.yandex.ru/>).

Инсталлируйте программу Skype, воспользовавшись ее адресом в сети Интернет <http://www.skype.com/intl/ru/get-skype>. После установки программа становится доступной после выполнения команды **Пуск/Все программы/ Skype/Skype**. В окне регистрации введите свой логин и пароль. Обратите внимание на то, что если вы установите флажок в пункте **Автоматическая авторизация при запуске Skype**, то вам не придется каждый раз вводить свои данные.

Добавьте своих друзей, родственников в список контактов, воспользовавшись командой **Контакты/Добавить контакт**. Вам нужно ввести фамилию, имя знакомого, его контактный телефон, адрес электронной почты. В результате ваши контакты будут располагаться в группе **Контакты** и будут видны при каждом запуске программы.

Выполним настройку web-камеры. Последовательно нажмем **Инструменты/Настройки/Настройки видео**. Появится окно, представленное на рис. 18.

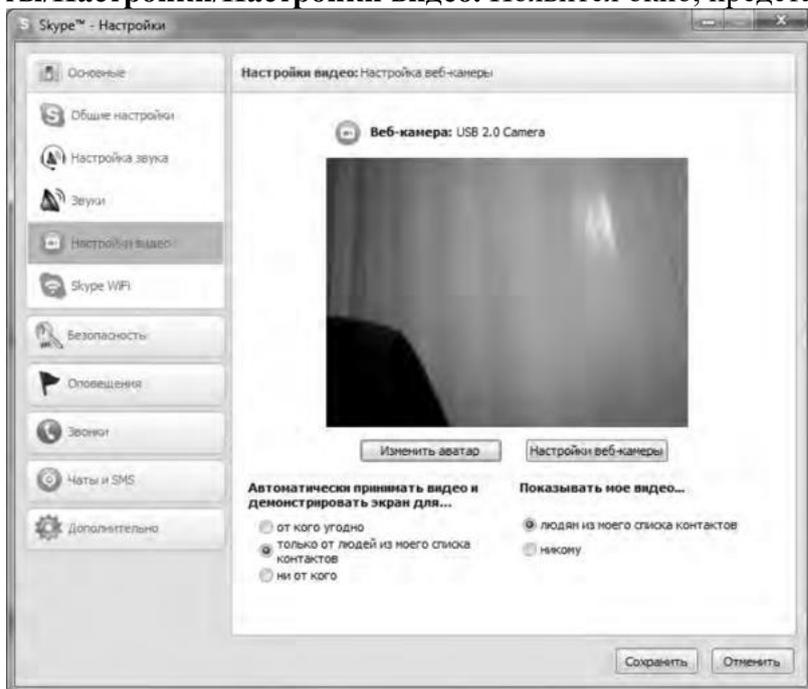


Рис.18. Окно Настройки

Если вы видите изображение - камера настроена и готова к работе. В противном случае, Skype выведет об этом текстовое сообщение. Теперь перейдем в меню **Настройка звука**. Проверьте, что поставлен флажок в опции **Разрешить автоматическую настройку микрофона**. Скажите несколько слов вслух, уровень громкости звука в опции **Громкость** должен изменяться. Окончательно проверить сделанные настройки можно с помощью контрольного звонка. Для этого, находясь в меню **Настройка звука**, выберите пункт **Сделать контрольный звонок в Skype**. В ходе контрольного звонка вы сможете сделать запись своего голоса в течение десяти секунд, а затем прослушать его. Если этот эксперимент окончится удачно, значит, все настройки выполнены правильно и программа готова к работе.

Теперь, когда мы завершили работу с настройками программы, можно попробовать сделать видеозвонок. Для этого необходимо совершить следующие действия:

1. Войти в программу Skype.
2. В группе **Контакты** щелчком мыши выбрать абонента. Во время звонка он должен быть в сети, о чем будет свидетельствовать соответствующий значок в программе Skype.
3. Нажать кнопку **Видеозвонок**.

Через несколько секунд соединение будет установлено и вы можете начать разговор, в процессе которого вы будете видеть и слышать своего собеседника. Подобная ситуация представлена на рис. 19.



Рис. 19 Сеанс связи установлен

Если во время разговоров у вас возникают неполадки со звуком, такие как сильный фоновый шум, эхо, задержка звука, «механический» звук или пропадание слов, следует убедиться в следующем:

1. Использует ли собеседник последнюю версию программы Skype? Информацию о версии программы можно получить, выполнив команду **По- мощь/О Skype**.
2. Нет ли рядом с микрофоном источников шума?
3. Не расположен ли микрофон рядом с динамиками?
4. Достаточно ли высокая скорость соединения?

Кроме того, когда программа Skype обнаруживает неполадки во время звонка, на экране появляется сообщение с рекомендациями, которые помогут вам повысить качество связи. Необходимо выполнить эти рекомендации.

Итак, вы получили теоретические сведения и практические навыки работы с организацией видео web-сессий, которые, несомненно, будут востребованы в вашей повседневной жизни.

Задания:

а) Зарегистрируйтесь на сервисе Forum2x2. Создайте форум своего учебного заведения, выбрав одну из четырех версий создания форумов. Выполните советы для успешного начала работы своего форума, приведенные в параграфе 5.4. После завершения работы отправьте на электронную почту преподавателя ссылку на созданный вами форум.

б) Установите на свой компьютер программу Skype. Сделайте видеозвонок вашему преподавателю (по предварительной договоренности).

2. Проведите диагностику стиля делового общения.

Инструкция. С помощью этого теста вы можете оценить свой стиль делового общения. Вам предложено 80 утверждений. Из каждой пары выберите одно — то, которое, как вы считаете, наиболее соответствует вашему поведению. Обратите внимание па то, что ни одна пара не должна быть пропущена. Тест построен таким образом, что ни одно из приведенных ниже утверждений не является ошибочным.

1. Я люблю действовать.
2. Я работаю над решением проблем систематическим образом.
3. Я считаю, что работа в командах более эффективна, чем на индивидуальной основе.
4. Мне очень нравятся различные нововведения.
5. Я больше интересуюсь будущим, чем прошлым.
6. Я очень люблю работать с людьми.
7. Я люблю принимать участие в хорошо организованных встречах.
8. Для меня очень важными являются окончательные сроки.

9. Я против откладываний и проволочек.
10. Я считаю, что новые идеи должны быть проверены прежде, чем они будут применяться на практике.
11. Я очень люблю взаимодействовать с другими людьми. Это меня стимулирует и вдохновляет.
12. Я всегда стараюсь искать новые возможности.
13. Я сам люблю устанавливать цели, планы и т.п.
14. Если я что-либо начинаю, то доделываю это до конца.
15. Обычно и стараюсь понять эмоциональные реакции других.
16. Я создаю проблемы другим людям.
17. Я надеюсь получить реакцию других на свое поведение.
18. Я нахожу, что действия, основанные на принципе «шаг за шагом», являются очень эффективными.
19. Я думаю, что хорошо могу понимать поведение и мысли других.
20. Я люблю творческое решение проблем.
21. Я все время строю планы на будущее.
22. Я восприимчив к нуждам других.
23. Хорошее планирование — ключ к успеху.
24. Меня раздражает слишком подробный анализ.
25. Я остаюсь невозмутимым, если на меня оказывают давление.
26. Я очень ценю опыт.
27. Я прислушиваюсь к мнению других.
28. Говорят, что я быстро соображаю.
29. Сотрудничество является для меня ключевым словом.
30. Я использую логические методы для анализа альтернатив.
31. Я люблю, когда одновременно у меня идут разные проекты.
32. Я постоянно задаю себе вопросы.
33. Делая что-либо, я тем самым учусь.
34. Полагаю, что я руководствуюсь рассудком, а не эмоциями.
35. Я могу предсказать, как другие будут вести себя в той или иной ситуации.
36. Я не люблю вдаваться в детали.
37. Анализ всегда должен предшествовать действиям.
38. Я способен оценить климат в группе.
39. У меня есть склонность не заканчивать начатые дела.
40. Я воспринимаю себя как решительного человека.
41. Я ищу такие дела, которые бросают мне вызов.
42. Я основываю свои действия на наблюдениях и фактах.
43. Я могу открыто выразить свои чувства.
44. Я люблю формулировать и определять контуры новых проектов.
45. Я очень люблю читать.
46. Я воспринимаю себя как человека, способного интенсифицировать, организовать деятельность других.
47. Я не люблю заниматься одновременно несколькими вопросами.
48. Я люблю достигать поставленных целей.
49. Мне нравится узнавать что-либо о других людях.
50. Я люблю разнообразие.
51. Факты говорят сами за себя.
52. Я использую свое воображение, насколько это возможно.
53. Меня раздражает длительная, кропотливая работа.
54. Мой мозг никогда не перестает работать.
55. Важному решению предшествует подготовительная работа.
56. Я глубоко уверен в том, что люди нуждаются друг в друге, чтобы завершить работу.

57. Я обычно принимаю решение, особо не задумываясь.
58. Эмоции только создают проблемы.
59. Я люблю быть таким же, как другие.
60. Я не могу быстро прибавить пятнадцать к семнадцати.
61. Я примеряю свои новые идеи к людям.
62. Я верю в научный подход.
63. Я люблю, когда дело сделано.
64. Хорошие отношения необходимы.
65. Я импульсивен.
66. Я нормально воспринимаю различия в людях.
67. Общение с другими людьми значимо само по себе.
68. Люблю, когда меня интеллектуально стимулируют.
69. Я люблю организовывать что-либо.
70. Я часто перескакиваю с одного дела на другое.
71. Общение и работа совместно с другими людьми являются творческим процессом.
72. Самоактуализация является крайне важной для меня.
73. Мне очень нравится играть идеями.
74. Я не люблю попусту терять время.
75. Я люблю делать то, что у меня получается.
76. Взаимодействуя с другими, я учусь.
77. Абстракции интересны для меня.
78. Мне нравятся детали.
79. Я люблю кратко подвести итоги, прежде чем прийти к какому-либо умозаключению.
80. Я достаточно уверен в себе.

Обработка результатов.

Обведите те номера, на которые вы ответили положительно, и отметьте их в приведенной ниже таблице. Посчитайте количество баллов по каждому стилю (один положительный ответ равен 1 баллу). Тот стиль, по которому вы набрали наибольшее количество баллов (по одному стилю не может быть более 20 баллов), наиболее предпочтителен для вас. Если вы набрали одинаковое количество баллов по двум стилям, значит, они оба присущи вам.

Ключ

Стиль 1: 1, 8, 9, 13, 17, 24, 26, 31, 33, 40, 41, 48, 50, 53, 57, 63, 65, 70, 74, 79.

Стиль 2: 2, 7, 10, 14, 18, 23, 25, 30, 34, 37, 42, 47, 51, 55, 58, 62, 66, 69, 75, 78.

Стиль 3: 3, 6, 11, 15, 19, 22, 27, 29, 35, 38, 43, 46, 49, 56, 59, 64, 67, 71, 76, 80.

Стиль 4: 4, 5, 12, 16, 20, 21, 28, 32, 36, 39, 44, 45, 52, 54, 60, 61, 68, 72, 73, 77.

Интерпретация результатов

Стиль 1 — ориентация на действие. Характерно обсуждение результатов, конкретных вопросов, поведения, ответственности, опыта, достижений, решений. Люди, владеющие этим стилем, прагматичны, прямолинейны, решительны, легко переключаются с одного вопроса на другой.

Стиль 2 — ориентация на процесс. Характерно обсуждение фактов, процедурных вопросов, планирования, организации, контролирования, деталей. Человек, владеющий этим стилем, ориентирован на систематичность, последовательность, тщательность. Он честен, многословен и мало эмоционален.

Стиль 3 ориентация на людей. Характерно обсуждение человеческих нужд, мотивов, чувств, «духа работы в команде», понимания, сотрудничества. Люди этого стиля эмоциональны, чувствительны, умеют сопереживать окружающим.

Стиль 4 — ориентация на перспективу, на будущее. Людям этого стиля присуще обсуждение концепций, больших планов, нововведений, различных вопросов, новых методов, альтернатив. Они обладают хорошим воображением, полны идей, но мало реалистичны и порой их сложно понять.

Задания:

- а) На основе самодиагностики определите стиль делового общения
- б) Дайте обоснование рекомендаций по совершенствованию делового общения.

ПОДГОТОВКА РЕФЕРАТА

Общая характеристика реферата

Написание реферата практикуется в учебном процессе в целях приобретения магистрантом необходимой профессиональной подготовки, развития умения и навыков самостоятельного научного поиска: изучения литературы по выбранной теме, анализа различных источников и точек зрения, обобщения материала, выделения главного, формулирования выводов и т. п. С помощью реферата магистрант может глубже постигать наиболее сложные проблемы дисциплины, учиться лаконично излагать свои мысли, правильно оформлять работу, докладывать результаты своего труда.

Реферат является первой ступенью на пути освоения навыков проведения научно-исследовательской работы. В «Толковом словаре русского языка» дается следующее определение: «**реферат** – краткое изложение содержания книги, статьи, исследования, а также доклад с таким изложением».

Различают два вида реферата:

- *репродуктивный* – воспроизводит содержание первичного текста в форме реферата-конспекта или реферата-резюме. В реферате-конспекте содержится фактическая информация в обобщённом виде, иллюстрированный материал, различные сведения о методах исследования, результатах исследования и возможностях их применения. В реферате-резюме содержатся только основные положения данной темы;

- *продуктивный* – содержит творческое или критическое осмысление реферируемого источника и оформляются в форме реферата-доклада или реферата-обзора. В реферате-докладе, наряду с анализом информации первоисточника, дается объективная оценка проблемы, и он имеет развёрнутый характер. Реферат-обзор составляется на основе нескольких источников и в нем сопоставляются различные точки зрения по исследуемой проблеме.

Магистрант для изложения материала должен выбрать продуктивный вид реферата.

Выбор темы реферата

Магистранту предоставляется право выбора темы реферата из рекомендованного преподавателем дисциплины списка. Выбор темы должен быть осознанным и обоснованным с точки зрения познавательных интересов автора, а также полноты освещения темы в имеющейся научной литературе.

Если интересующая тема отсутствует в рекомендованном списке, то по согласованию с преподавателем магистранту предоставляется право самостоятельно предложить тему реферата, раскрывающую содержание изучаемой дисциплины. Тема не должна быть слишком общей и глобальной, так как небольшой объем работы (до 20-25 страниц без учёта приложений) не позволит раскрыть ее.

Начинать знакомство с избранной темой лучше всего с чтения обобщающих работ по данной проблеме, постепенно переходя к узкоспециальной литературе. При этом следует сразу же составлять

библиографические выходные данные используемых источников (автор, название, место и год издания, издательство, страницы).

На основе анализа прочитанного и просмотренного материала по данной теме следует составить тезисы по основным смысловым блокам, с пометками, собственными суждениями и оценками. Предварительно подобранный в литературных источниках материал может превышать необходимый объем реферата.

Формулирование цели и составление плана реферата

Выбрав тему реферата и изучив литературу, необходимо сформулировать цель работы и составить план реферата.

Цель – это осознаваемый образ предвосхищаемого результата. Возможно, формулировка цели в ходе работы будет меняться, но изначально следует ее обозначить, чтобы ориентироваться на нее в ходе исследования. Формулирование цели реферата рекомендуется осуществлять при помощи глаголов: исследовать, изучить, проанализировать, систематизировать, осветить, изложить (представления, сведения), создать, рассмотреть, обобщить и т. д.

Определяясь с целью дальнейшей работы, параллельно необходимо думать над составлением плана, при этом четко соотносить цель и план работы. Правильно построенный план помогает систематизировать материал и обеспечить последовательность его изложения.

Наиболее традиционной является следующая **структура реферата**:

Титульный лист.

Оглавление (план, содержание).

Введение.

1. (полное наименование главы).

1.1. (полное название параграфа, пункта);

1.2. (полное название параграфа, пункта).

Основная часть

2. (полное наименование главы).

2.1. (полное название параграфа, пункта);

2.2. (полное название параграфа, пункта).

Заключение (выводы).

Библиография (список использованной литературы).

Приложения (по усмотрению автора).

Титульный лист оформляется в соответствии с Приложением.

Оглавление (план, содержание) включает названия всех глав и параграфов (пунктов плана) реферата и номера страниц, указывающие их начало в тексте реферата.

Введение. В этой части реферата обосновывается актуальность выбранной темы, формулируются цель и задачи работы, указываются используемые материалы и дается их краткая характеристика с точки зрения полноты освещения избранной темы. Объем введения не должен превышать 1-1,5 страницы.

Основная часть реферата может быть представлена двумя или тремя главами, которые могут включать 2-3 параграфа (пункта).

Здесь достаточно полно и логично излагаются главные положения в используемых источниках, раскрываются все пункты плана с сохранением связи между ними и последовательности перехода от одного к другому.

Автор должен следить за тем, чтобы изложение материала точно соответствовало цели и названию главы (параграфа). Материал в реферате рекомендуется излагать своими словами, не допуская дословного переписывания из литературных источников. В тексте обязательны ссылки на первоисточники, т. е. на тех авторов, у которых взят данный материал в виде мысли, идеи, вывода, числовых данных, таблиц, графиков, иллюстраций и пр.

Работа должна быть написана грамотным литературным языком. Сокращение слов в тексте не допускается, кроме общеизвестных сокращений и аббревиатуры. Каждый раздел рекомендуется заканчивать кратким выводом.

Заключение (выводы). В этой части обобщается изложенный в основной части материал, формулируются общие выводы, указывается, что нового лично для себя вынес автор реферата из работы над ним. Выводы делаются с учетом опубликованных в литературе различных точек зрения по проблеме рассматриваемой в реферате, сопоставления их и личного мнения автора реферата. Заключение по объему не должно превышать 1,5-2 страниц.

Библиография (список использованной литературы) – здесь указывается реально использованная для написания реферата литература, периодические издания и электронные источники информации. Список составляется согласно правилам библиографического описания.

Приложения могут включать графики, таблицы, расчеты.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ РЕФЕРАТА

Общие требования к оформлению реферата

Рефераты по дисциплинам магистратуры направления подготовки 38.04.02 – «Менеджмент», как правило, требуют изучения и анализа значительного объема статистического материала, формул, графиков и т. п. В силу этого особое значение приобретает правильное оформление результатов проделанной работы.

Текст реферата должен быть подготовлен в печатном виде. Исправления и пометки не допускаются. Текст работы оформляется на листах формата А4, на одной стороне листа, с полями: левое – 25 мм, верхнее – 20 мм, правое – 15 мм и нижнее – 25 мм. При компьютерном наборе шрифт должен быть таким: тип шрифта Times New Roman, кегль 14, междустрочный интервал 1,5.

Рекомендуемый объем реферата – не менее 20 страниц. Титульный лист реферата оформляется магистрантом по образцу, данному в приложении 1.

Текст реферата должен быть разбит на разделы: главы, параграфы и т. д. Очередной раздел нужно начинать с нового листа.

Все страницы реферата должны быть пронумерованы. Номер страницы ставится снизу страницы, по центру. Первой страницей является титульный лист, но на ней номер страницы не ставится.

Таблицы

Таблицы по содержанию делятся на аналитические и неаналитические. Аналитические таблицы являются результатом обработки и анализа цифровых показателей. Как правило, после таких таблиц делается обобщение, которое вводится в текст словами: «таблица позволяет сделать вывод о том, что...», «таблица позволяет заключить, что...» и т. п.

В неаналитических таблицах обычно помещаются необработанные статистические данные, необходимые лишь для информации и констатации фактов.

Таблицы размещают после первого упоминания о них в тексте таким образом, чтобы их можно было читать без поворота работы или с поворотом по часовой стрелке.

Каждая таблица должна иметь нумерационный и тематический заголовок. Тематический заголовок располагается по центру таблицы, после нумерационного, размещённого в правой стороне листа и включающего надпись «Таблица» с указанием арабскими цифрами номера таблицы. Нумерация таблиц сквозная в пределах каждой главы. Номер таблицы состоит из двух цифр: первая указывает на номер главы, вторая – на номер таблицы в главе по порядку (например: «Таблица 2.2» – это значит, что представленная таблица вторая во второй главе).

Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим. В одной графе количество десятичных знаков должно быть одинаковым. Если данные отсутствуют, то в графах ставят знак тире. Округление числовых значений величин до первого, второго и т. д. десятичного знака для различных значений одного и того же наименования показателя должно быть одинаковым.

Таблицу с большим количеством строк допускается переносить на другую страницу, при этом заголовок таблицы помещают только над ее первой частью, а над переносимой частью пишут «Продолжение таблицы» или «Окончание таблицы». Если в работе несколько таблиц, то после слов «Продолжение» или «Окончание» указывают номер таблицы, а само слово «таблица» пишут сокращенно, например: «Продолжение табл. 1.1», «Окончание табл. 1.1».

На все таблицы в тексте курсовой работы должны быть даны ссылки с указанием их порядкового номера, например: «...в табл. 2.2».

Формулы

Формулы – это комбинации математических знаков, выражающие какие-либо предложения.

Формулы, приводимые в реферате, должны быть наглядными, а обозначения, применяемые в них, соответствовать стандартам.

Пояснения значений символов и числовых коэффициентов следует приводить непосредственно под формулой, в той последовательности, в какой они даны в формуле. Значение каждого символа и числового коэффициента дается с новой строки. Первую строку объяснения начинают со слова «где» без двоеточия после него.

Формулы и уравнения следует выделять из текста свободными строками. Если уравнение не умещается в одну строку, оно должно быть перенесено после знака равенства (=) или после знака (+), минус (–), умножения (x) и деления (:).

Формулы нумеруют арабскими цифрами в пределах всей курсовой работы (реферата) или главы. В пределах реферата используют нумерацию формул одинарную, в пределах главы – двойную. Номер указывают с правой стороны листа на уровне формулы в круглых скобках.

В тексте ссылки на формулы приводятся с указанием их порядковых номеров, например: «...в формуле (2.2)» (второй формуле второй главы).

Иллюстрации

Иллюстрации позволяют наглядно представить явление или предмет такими, какими мы их зрительно воспринимаем, но без лишних деталей и подробностей.

Основными видами иллюстраций являются схемы, диаграммы и графики.

Схема – это изображение, передающее обычно с помощью условных обозначений и без соблюдения масштаба основную идею какого-либо устройства, предмета, сооружения или процесса и показывающее взаимосвязь их главных элементов.

Диаграмма – один из способов изображения зависимости между величинами. Наибольшее распространение получили линейные, столбиковые и секторные диаграммы.

Для построения линейных диаграмм используется координатное поле. По горизонтальной оси в изображенном масштабе откладывается время или факториальные признаки, на вертикальной – показатели на определенный момент (период) времени или размеры результативного независимого признака. Вершины ординат соединяются отрезками – в результате получается ломаная линия.

На столбиковых диаграммах данные изображаются в виде прямоугольников (столбиков) одинаковой ширины, расположенных вертикально или горизонтально. Длина (высота) прямоугольников пропорциональна изображенным ими величинам.

Секторная диаграмма представляет собой круг, разделенный на секторы, величины которых пропорциональны величинам частей изображаемого явления.

График – это результат обработки числовых данных. Он представляет собой условные изображения величин и их соотношений через геометрические фигуры, точки и линии.

Количество иллюстраций в работе должно быть достаточным для пояснения излагаемого текста.

Иллюстрации обозначаются словом «Рис.» и располагаются после первой ссылки на них в тексте так, чтобы их было удобно рассматривать без поворота

работы или с поворотом по часовой стрелке. Иллюстрации должны иметь номер и наименование, расположенные по центру, под ней. Иллюстрации нумеруются в пределах главы арабскими цифрами, например: «Рис. 1.1» (первый рисунок первой главы). Ссылки на иллюстрации в тексте реферата приводят с указанием их порядкового номера, например: «...на рис. 1.1».

При необходимости иллюстрации снабжаются поясняющими данными (подрисуночный текст).

Приложения

Приложение – это часть основного текста, которая имеет дополнительное (обычно справочное) значение, но, тем не менее, необходима для более полного освещения темы. По форме они могут представлять собой текст, таблицы, графики, карты. В приложении помещают вспомогательные материалы по рассматриваемой теме: инструкции, методики, положения, результаты промежуточных расчетов, типовые проекты, имеющие значительный объем, затрудняющий чтение и целостное восприятие текста. В этом случае в тексте приводятся основные выводы (результаты) и делается ссылка на приложение, содержащее соответствующую информацию. Каждое приложение должно начинаться с новой страницы. В правом верхнем углу листа пишут слово «Приложение» и указывают номер приложения. Если в реферате больше одного приложения, их нумеруют последовательно арабскими цифрами, например: «Приложение 1», «Приложение 2» и т. д.

Каждое приложение должно иметь заголовок, который помещают ниже слова «Приложение» над текстом приложения, по центру.

При ссылке на приложение в тексте реферата пишут сокращенно строчными буквами «прил.» и указывают номер приложения, например: «...в прил. 1».

Приложения оформляются как продолжение текстовой части реферата со сквозной нумерацией листов. Число страниц в приложении не лимитируется и не включается в общий объем страниц реферата.

Библиографический список

Библиографический список должен содержать перечень и описание только тех источников, которые были использованы при написании реферата.

В библиографическом списке должны быть представлены монографические издания отечественных и зарубежных авторов, материалы профессиональной периодической печати (экономических журналов, газет и еженедельников), законодательные и др. нормативно-правовые акты. При составлении списка необходимо обратить внимание на достижение оптимального соотношения между монографическими изданиями, характеризующими глубину теоретической подготовки автора, и периодикой, демонстрирующей владение современными экономическими данными.

Наиболее распространенным способом расположения наименований литературных источников является алфавитный. Работы одного автора перечисляются в алфавитном порядке их названий. Исследования на

иностранных языках помещаются в порядке латинского алфавита после исследований на русском языке.

Ниже приводятся примеры библиографических описаний использованных источников.

Статья одного, двух или трех авторов из журнала

Зотова Л. А., Еременко О. В. Инновации как объект государственного регулирования // *Экономист*. 2010. № 7. С. 17–19.

Статья из журнала, написанная более чем тремя авторами

Валютный курс и экономический рост / С. Ф. Алексахенко, А. А. Клепач, О. Ю. Осипова [и др.] // *Вопросы экономики*. 2010. № 8. С. 18–22.

Книга, написанная одним, двумя или тремя авторами

Иохин В. Я. Экономическая теория: учебник. М.: Юристъ, 2009. 178 с.

Книга, написанная более чем тремя авторами

Экономическая теория: учебник / В. Д. Камаев [и др.]. М.: ВЛАДОС, 2011. 143 с.

Сборники

Актуальные проблемы экономики и управления: сборник научных статей. Екатеринбург: УГГУ, 2010. Вып. 9. 146 с.

Статья из сборника

Данилов А. Г. Система ценообразования промышленного предприятия // *Актуальные проблемы экономики и управления: сб. научных статей*. Екатеринбург: УГГУ, 2010. Вып. 9. С. 107–113.

Статья из газеты

Крашаков А. С. Будет ли обвал рубля // *Аргументы и факты*. 2011. № 9. С. 3.

Библиографические ссылки

Библиографические ссылки требуется приводить при цитировании, заимствовании материалов из других источников, упоминании или анализе работ того или иного автора, а также при необходимости адресовать читателя к трудам, в которых рассматривался данный вопрос.

Ссылки должны быть затекстовыми, с указанием номера соответствующего источника (на который автор ссылается в работе) в соответствии с библиографическим списком и соответствующей страницы.

Пример оформления затекстовой ссылки

Ссылка в тексте: «При оценке стоимости земли необходимо учесть все возможности ее производственного использования» [17, С. 191].

В списке использованных источников:

17. *Борисов Е. Ф.* Основы экономики. М.: Юристъ, 2008. 308 с.

ОРГАНИЗАЦИЯ ЗАЩИТЫ РЕФЕРАТА

Необходимо заранее подготовить тезисы выступления (план-конспект).

Порядок защиты реферата.

1. Краткое сообщение, характеризующее цель и задачи работы, ее актуальность, полученные результаты, вывод и предложения.
2. Ответы магистранта на вопросы преподавателя.
3. Отзыв руководителя-консультанта о ходе выполнения работы.

Советы магистранту:

•Готовясь к защите реферата, вы должны вспомнить материал максимально подробно, и это должно найти отражение в схеме вашего ответа. Но тут же необходимо выделить главное, что наиболее важно для понимания материала в целом, иначе вы сможете проговорить все 15-20 минут и не раскрыть существа вопроса. Особенно строго следует отбирать примеры и иллюстрации.

•Вступление должно быть очень кратким – 1-2 фразы (если вы хотите подчеркнуть при этом важность и сложность данного вопроса, то не говорите, что он сложен и важен, а покажите его сложность и важность).

•Целесообразнее вначале показать свою схему раскрытия вопроса, а уж потом ее детализировать.

•Рассказывать будет легче, если вы представите себе, что объясняете материал очень способному и хорошо подготовленному человеку, который не знает именно этого раздела, и что при этом вам обязательно нужно доказать важность данного раздела и заинтересовать в его освоении.

•Строго следите за точностью своих выражений и правильностью употребления терминов.

•Не пытайтесь рассказать побольше за счет ускорения темпа, но и не мямлите.

•Не демонстрируйте излишнего волнения и не напрашивайтесь на сочувствие.

•Будьте особенно внимательны ко всем вопросам преподавателя, к малейшим его замечаниям. И уж ни в коем случае его не перебивайте!

•Не бойтесь дополнительных вопросов – чаще всего преподаватель использует их как один из способов помочь вам или сэкономить время. Если вас прервали, а при оценке ставят в вину пропуск важной части материала, не возмущайтесь, а покажите план своего ответа, где эта часть стоит несколько позже того, на чем вы были прерваны.

•Прежде чем отвечать на дополнительный вопрос, необходимо сначала правильно его понять. Для этого нужно хотя бы немного подумать, иногда переспросить, уточнить: правильно ли вы поняли поставленный вопрос. И при ответе следует соблюдать тот же принцип экономности мышления, а не высказывать без разбора все, что вы можете сказать.

•Будьте доброжелательны и тактичны, даже если к ответу вы не готовы (это вина не преподавателя, а ваша).

ТЕМЫ РЕФЕРАТА

1. Общение как социально-психологическая категория.
2. Коммуникативная культура в деловом общении.
3. Условия общения и причины коммуникативных неудач.
4. Роль невербальных компонентов в речевом общении.
5. Речевой этикет, его основные функции и правила.
6. Причины отступлений от норм в речи, типы речевых ошибок, пути их устранения и предупреждения.
7. Деловая беседа (цели, задачи, виды, структура).
8. Особенности телефонного разговора.
9. Новые тенденции в практике русского делового письма.
10. Культура дискусивно-полемиической речи. Виды споров, приемы и уловки в споре
11. Основные правила эффективного общения.
12. Личность как субъект общения. Коммуникативная компетентность личности.
13. Конфликтное поведение и причины его возникновения в деструктивном взаимодействии.
14. Деловое общение и управление им.
15. Отношения сотрудничества и конфликта в представлениях российских работников.
16. Реформы в России и проблемы общения молодого поколения и работодателей.
17. Культура речи в деловом общении.
18. Содержание закона конгруэнтности и его роль в деловом общении.
19. Этика использования средств выразительности деловой речи.
20. Особенности речевого поведения.
21. Культура устной и письменной речи делового человека в современной России.
22. Вербальные конфликтогены в практике современного российского общества.
23. Этические нормы телефонного разговора.
24. Основные тенденции развития Российской деловой культуры.
25. Характеристика манипуляций в общении.
26. Приемы, стимулирующие общение и создание доверительных отношений.
27. Правила подготовки публичного выступления.
28. Правила подготовки и проведения деловой беседы.
29. Типология конфликтных личностей и способы общения с ними.
30. Этикет и имидж делового человека.

ПОДГОТОВКА К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

При подготовке к зачету по дисциплине «Средства коммуникации в учебной и профессиональной деятельности» обучающемуся рекомендуется:

1. повторить пройденный материал и ответить на вопросы, используя конспект и материалы лекций. Если по каким-либо вопросам у студента недостаточно информации в лекционных материалах, то необходимо получить информацию из раздаточных материалов и/или учебников (литературы), рекомендованных для изучения дисциплины «Средства коммуникации в учебной и профессиональной деятельности».

Целесообразно также дополнить конспект лекций наиболее существенными и важными тезисами для рассматриваемого вопроса;

2. при изучении основных и дополнительных источников информации в рамках выполнения заданий на зачете особое внимание необходимо уделять схемам, рисункам, графикам и другим иллюстрациям, так как подобные графические материалы, как правило, в наглядной форме отражают главное содержание изучаемого вопроса;

3. при изучении основных и дополнительных источников информации в рамках выполнения заданий на зачете (в случаях, когда отсутствует иллюстративный материал) особое внимание необходимо обращать на наличие в тексте словосочетаний вида «во-первых», «во-вторых» и т.д., а также дефисов и перечислений (цифровых или буквенных), так как эти признаки, как правило, позволяют структурировать ответ на предложенное задание.

Подобную текстовую структуризацию материала слушатель может трансформировать в рисунки, схемы и т. п. для более краткого, наглядного и удобного восприятия (иллюстрации целесообразно отразить в конспекте лекций – это позволит оперативно и быстро найти, в случае необходимости, соответствующую информацию);

4. следует также обращать внимание при изучении материала для подготовки к зачету на словосочетания вида «таким образом», «подводя итог сказанному» и т.п., так как это признаки выражения главных мыслей и выводов по изучаемому вопросу (пункту, разделу). В отдельных случаях выводы по теме (разделу, главе) позволяют полностью построить (восстановить, воссоздать) ответ на поставленный вопрос (задание), так как содержат в себе основные мысли и тезисы для ответа.

СОДЕРЖАНИЕ

Введение	3
1 Методические рекомендации по решению практико-ориентированных заданий	5
2 Методические указания по подготовке к опросу	9
3 Методические рекомендации по подготовке к практическим занятиям	11
4 Методические рекомендации по подготовке семинарским занятиям	13
5 Методические рекомендации по подготовке к сдаче экзаменов и зачетов	14
Заключение	17
Список использованных источников	18

ВВЕДЕНИЕ

Самостоятельная работа студентов может рассматриваться как организационная форма обучения - система педагогических условий, обеспечивающих управление учебной деятельностью студентов по освоению знаний и умений в области учебной и научной деятельности без посторонней помощи.

Самостоятельная работа студентов проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений студентов;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирования практических (общеучебных и профессиональных) умений и навыков;
- развития исследовательских умений;
- получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная образовательной программой, учебным планом и учебно-методическими материалами, раскрывающими и конкретизирующими их содержание, осуществляется студентами инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует студентам источники и учебно-методические пособия для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т. п.

Подразумевается несколько категорий видов самостоятельной работы студентов, значительная часть которых нашла отражения в данных методических рекомендациях:

- работа с источниками литературы и официальными документами (*использование библиотечно-информационной системы*);
- выполнение заданий для самостоятельной работы в рамках учебных дисциплин (*рефераты, эссе, домашние задания, решения практико-ориентированных заданий*);

- реализация элементов научно-педагогической практики (*разработка методических материалов, тестов, тематических портфолио*);
- реализация элементов научно-исследовательской практики (*подготовка текстов докладов, участие в исследованиях*).

Особенностью организации самостоятельной работы студентов является необходимость не только подготовиться к сдаче зачета, но и собрать, обобщить, систематизировать, проанализировать информацию по темам дисциплины.

Технология организации самостоятельной работы студентов включает использование информационных и материально-технических ресурсов образовательного учреждения.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

В качестве форм и методов контроля внеаудиторной самостоятельной работы студентов могут быть использованы обмен информационными файлами, семинарские занятия, тестирование, опрос, доклад, реферат, самоотчеты, контрольные работы, защита творческих работ и электронных презентаций и др.

Контроль результатов внеаудиторной самостоятельной работы студентов осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине.

1. Методические рекомендации по решению практико-ориентированных заданий

Практико-ориентированные задания - метод анализа ситуаций. Суть его заключается в том, что студентам предлагают осмыслить реальную жизненную ситуацию, описание которой одновременно отражает не только какую-либо практическую проблему, но и актуализирует определенный комплекс знаний, который необходимо усвоить при разрешении данной проблемы. При этом сама проблема не имеет однозначных решений.

Использование метода практико-ориентированного задания как образовательной технологии профессионально-ориентированного обучения представляет собой сложный процесс, плохо поддающийся алгоритмизации¹. Формально можно выделить следующие этапы:

- ознакомление студентов с текстом;
- анализ практико-ориентированного задания;
- организация обсуждения практико-ориентированного задания, дискуссии, презентации;
- оценивание участников дискуссии;
- подведение итогов дискуссии.

Ознакомление студентов с текстом практико-ориентированного задания и последующий анализ практико-ориентированного задания чаще всего осуществляются за несколько дней до его обсуждения и реализуются как самостоятельная работа студентов; при этом время, отводимое на подготовку, определяется видом практико-ориентированного задания, его объемом и сложностью.

Общая схема работы с практико-ориентированным заданием на данном этапе может быть представлена следующим образом: в первую очередь следует выявить ключевые проблемы практико-ориентированного задания и понять, какие именно из представленных данных важны для решения; войти в ситуационный контекст практико-ориентированного задания, определить, кто его главные действующие лица, отобрать факты и понятия, необходимые для анализа, понять, какие трудности могут возникнуть при решении задачи; следующим этапом является выбор метода исследования.

Знакомство с небольшими практико-ориентированными заданиями и их обсуждение может быть организовано непосредственно на занятиях. Принципиально важным в этом случае является то, чтобы часть теоретического курса, на которой базируется практико-ориентированное задание, была бы прочитана и проработана студентами.

Максимальная польза из работы над практико-ориентированными заданиями будет извлечена в том случае, если аспиранты при предварительном знакомстве с ними будут придерживаться систематического подхода к их анализу, основные шаги которого представлены ниже:

1. Выпишите из соответствующих разделов учебной дисциплины ключевые идеи, для того, чтобы освежить в памяти теоретические концепции и подходы, которые Вам предстоит использовать при анализе практико-ориентированного задания.
2. Бегло прочтите практико-ориентированное задание, чтобы составить о нем общее представление.
3. Внимательно прочтите вопросы к практико-ориентированному заданию и убедитесь в том, что Вы хорошо поняли, что Вас просят сделать.
4. Вновь прочтите текст практико-ориентированного задания, внимательно фиксируя все факторы или проблемы, имеющие отношение к поставленным вопросам.
5. Прикиньте, какие идеи и концепции соотносятся с проблемами, которые Вам предлагается рассмотреть при работе с практико-ориентированным заданием.

¹ Долгоруков А. Метод case-study как современная технология профессионально -ориентированного обучения [Электронный ресурс]. Режим доступа: <http://evolkov.net/case/case.study.html/>

Организация обсуждения практико-ориентированного задания предполагает формулирование перед студентами вопросов, включение их в дискуссию. Вопросы обычно подготавливаются заранее и предлагают студентам вместе с текстом практико-ориентированного задания. При разборе учебной ситуации преподаватель может занимать активную или пассивную позицию, иногда он «дирижирует» разбором, а иногда ограничивается подведением итогов дискуссии.

Организация обсуждения практико-ориентированных заданий обычно основывается на двух методах. Первый из них носит название традиционного Гарвардского метода - открытая дискуссия. Альтернативным методом является метод, связанный с индивидуальным или групповым опросом, в ходе которого аспиранты делают формальную устную оценку ситуации и предлагают анализ представленного практико-ориентированного задания, свои решения и рекомендации, т.е. делают презентацию. Этот метод позволяет некоторым студентам минимизировать их учебные усилия, поскольку каждый аспирант опрашивается один-два раза за занятие. Метод развивает у студентов коммуникативные навыки, учит их четко выражать свои мысли. Однако, этот метод менее динамичен, чем Гарвардский метод. В открытой дискуссии организация и контроль участников более сложен.

Дискуссия занимает центральное место в методе. Ее целесообразно использовать в том случае, когда аспиранты обладают значительной степенью зрелости и самостоятельности мышления, умеют аргументировать, доказывать и обосновывать свою точку зрения. Важнейшей характеристикой дискуссии является уровень ее компетентности, который складывается из компетентности ее участников. Неподготовленность студентов к дискуссии делает ее формальной, превращает в процесс вытаскивания ими информации у преподавателя, а не самостоятельное ее добывание.

Особое место в организации дискуссии при обсуждении и анализе практико-ориентированного задания принадлежит использованию метода генерации идей, получившего название «мозговой атаки» или «мозгового штурма».

Метод «мозговой атаки» или «мозгового штурма» был предложен в 30-х годах прошлого столетия А. Осборном как групповой метод решения проблем. К концу XX столетия этот метод приобрел особую популярность в практике управления и обучения не только как самостоятельный метод, но и как использование в процессе деятельности с целью усиления ее продуктивности. В процессе обучения «мозговая атака» выступает в качестве важнейшего средства развития творческой активности студентов. «Мозговая атака» включает в себя три фазы.

Первая фаза представляет собой вхождение в психологическую раскованность, отказ от стереотипности, страха показаться смешным и неудачником; достигается созданием благоприятной психологической обстановки и взаимного доверия, когда идеи теряют авторство, становятся общими. Основная задача этой фазы - успокоиться и расковаться.

Вторая фаза - это собственно атака; задача этой фазы - породить поток, лавину идей. «Мозговая атака» в этой фазе осуществляется по следующим принципам:

- есть идея, - говорю, нет идеи, - не молчу;
- поощряется самое необузданное ассоциирование, чем более дикой покажется идея, тем лучше;
- количество предложенных идей должно быть как можно большим;
- высказанные идеи разрешается заимствовать и как угодно комбинировать, а также видоизменять и улучшать;
- исключается критика, можно высказывать любые мысли без боязни, что их признают плохими, критикующих лишают слова;
- не имеют никакого значения социальные статусы участников; это абсолютная демократия и одновременно авторитаризм сумасшедшей идеи;
- все идеи записываются в протокольный список идей;

- время высказываний - не более 1-2 минут.

Третья фаза представляет собой творческий анализ идей с целью поиска конструктивного решения проблемы по следующим правилам:

- анализировать все идеи без дискриминации какой-либо из них;
- найти место идее в системе и найти систему под идею;
- не умножать сущностей без надобности;
- не должна нарушаться красота и изящество полученного результата;
- должно быть принципиально новое видение;
- ищи «жемчужину в навозе».

В методе мозговая атака применяется при возникновении у группы реальных затруднений в осмыслении ситуации, является средством повышения активности студентов. В этом смысле мозговая атака представляется не как инструмент поиска новых решений, хотя и такая ее роль не исключена, а как своеобразное «подталкивание» к познавательной активности.

Презентация, или представление результатов анализа практико-ориентированного задания, выступает очень важным аспектом метода *case-study*. Умение публично представить интеллектуальный продукт, хорошо его рекламировать, показать его достоинства и возможные направления эффективного использования, а также выстоять под шквалом критики, является очень ценным интегральным качеством современного специалиста. Презентация оттачивает многие глубинные качества личности: волю, убежденность, целенаправленность, достоинство и т.п.; она вырабатывает навыки публичного общения, формирования своего собственного имиджа.

Публичная (устная) презентация предполагает представление решений практико-ориентированного задания группе, она максимально вырабатывает навыки публичной деятельности и участия в дискуссии. Устная презентация обладает свойством кратковременного воздействия на студентов и, поэтому, трудна для восприятия и запоминания. Степень подготовленности выступающего проявляется в спровоцированной им дискуссии: для этого необязательно делать все заявления очевидными и неопровержимыми. Такая подача материала при анализе практико-ориентированного задания может послужить началом дискуссии. При устной презентации необходимо учитывать эмоциональный настрой выступающего: отношение и эмоции говорящего вносят существенный вклад в сообщение. Одним из преимуществ публичной (устной) презентации является ее гибкость. Оратор может откликаться на изменения окружающей обстановки, адаптировать свой стиль и материал, чувствуя настроение аудитории.

Непубличная презентация менее эффективна, но обучающая роль ее весьма велика. Чаще всего непубличная презентация выступает в виде подготовки отчета по выполнению задания, при этом стимулируются такие качества, как умение подготовить текст, точно и аккуратно составить отчет, не допустить ошибки в расчетах и т.д. Подготовка письменного анализа практико-ориентированного задания аналогична подготовке устного, с той разницей, что письменные отчеты-презентации обычно более структурированы и детализированы. Основное правило письменного анализа практико-ориентированного задания заключается в том, чтобы избегать простого повторения информации из текста, информация должна быть представлена в переработанном виде. Самым важным при этом является собственный анализ представленного материала, его соответствующая интерпретация и сделанные предложения. Письменный отчет - презентация может сдаваться по истечении некоторого времени после устной презентации, что позволяет студентам более тщательно проанализировать всю информацию, полученную в ходе дискуссии.

Как письменная, так и устная презентация результатов анализа практико-ориентированного задания может быть групповой и индивидуальной. Отчет может быть индивидуальным или групповым в зависимости от сложности и объема задания. Индивидуальная презентация формирует ответственность, собранность, волю;

групповая - аналитические способности, умение обобщать материал, системно видеть проект.

Оценивание участников дискуссии является важнейшей проблемой обучения посредством метода практико-ориентированного задания. При этом выделяются следующие требования к оцениванию:

- объективность - создание условий, в которых бы максимально точно выявлялись знания обучаемых, предъявление к ним единых требований, справедливое отношение к каждому;
- обоснованность оценок - их аргументация;
- систематичность - важнейший психологический фактор, организующий и дисциплинирующий студентов, формирующий настойчивость и устремленность в достижении цели;
- всесторонность и оптимальность.

Оценивание участников дискуссии предполагает оценивание не столько набора определенных знаний, сколько умения студентов анализировать конкретную ситуацию, принимать решение, логически мыслить.

Следует отметить, что оценивается содержательная активность студента в дискуссии или публичной (устной) презентации, которая включает в себя следующие составляющие:

- выступление, которое характеризует попытку серьезного предварительного анализа (правильность предложений, подготовленность, аргументированность и т.д.);
- обращение внимания на определенный круг вопросов, которые требуют углубленного обсуждения;
- владение категориальным аппаратом, стремление давать определения, выявлять содержание понятий;
- демонстрация умения логически мыслить, если точки зрения, высказанные раньше, подытоживаются и приводят к логическим выводам;
- предложение альтернатив, которые раньше оставались без внимания;
- предложение определенного плана действий или плана воплощения решения;
- определение существенных элементов, которые должны учитываться при анализе практико-ориентированного задания;
- заметное участие в обработке количественных данных, проведении расчетов;
- подведение итогов обсуждения.

При оценивании анализа практико-ориентированного задания, данного студентами при непубличной (письменной) презентации учитывается:

- формулировка и анализ большинства проблем, имеющих в практико-ориентированное задание;
- формулировка собственных выводов на основании информации о практико-ориентированное задание, которые отличаются от выводов других студентов;
- демонстрация адекватных аналитических методов для обработки информации;
- соответствие приведенных в итоге анализа аргументов ранее выявленным проблемам, сделанным выводам, оценкам и использованным аналитическим методам.

2. Методические указания по подготовке к опросу

Самостоятельная работа обучающихся включает подготовку к устному или письменному опросу на семинарских занятиях. Для этого обучающийся изучает лекции, основную и дополнительную литературу, публикации, информацию из Интернет-ресурсов. Темы и вопросы к семинарским занятиям, вопросы для самоконтроля приведены в методических указаниях по разделам и доводятся до обучающихся заранее.

Письменный опрос

В соответствии с технологической картой письменный опрос является одной из форм текущего контроля успеваемости студента по данной дисциплине. При подготовке к письменному опросу студент должен внимательно изучать лекции, основную и дополнительную литературу, публикации, информацию из Интернет-ресурсов. Темы и вопросы к семинарским занятиям, вопросы для самоконтроля приведены в методических указаниях по разделам и доводятся до обучающихся заранее.

При изучении материала студент должен убедиться, что хорошо понимает основную терминологию темы, умеет ее использовать в нужном контексте. Желательно составить краткий конспект ответа на предполагаемые вопросы письменной работы, чтобы убедиться в том, что студент владеет материалом и может аргументировано, логично и грамотно письменно изложить ответ на вопрос. Следует обратить особое внимание на написание профессиональных терминов, чтобы избежать грамматических ошибок в работе. При изучении новой для студента терминологии рекомендуется изготовить карточки, которые содержат новый термин и его расшифровку, что значительно облегчит работу над материалом.

Устный опрос

Целью устного собеседования являются обобщение и закрепление изученного курса. Студентам предлагаются для освещения сквозные концептуальные проблемы. При подготовке следует использовать лекционный материал и учебную литературу. Для более глубокого постижения курса и более основательной подготовки рекомендуется познакомиться с указанной дополнительной литературой. Готовясь к семинару, студент должен, прежде всего, ознакомиться с общим планом семинарского занятия. Следует внимательно прочесть свой конспект лекции по изучаемой теме и рекомендуемую к теме семинара литературу. С незнакомыми терминами и понятиями следует ознакомиться в предлагаемом глоссарии, словаре или энциклопедии².

Критерии качества устного ответа.

1. Правильность ответа по содержанию.
2. Полнота и глубина ответа.
3. Сознательность ответа (учитывается понимание излагаемого материала).
4. Логика изложения материала (учитывается умение строить целостный, последовательный рассказ, грамотно пользоваться профессиональной терминологией).
5. Рациональность использованных приемов и способов решения поставленной учебной задачи (учитывается умение использовать наиболее прогрессивные и эффективные способы достижения цели).
6. Своевременности и эффективность использования наглядных пособий и технических средств при ответе (учитывается грамотно и с пользой применять наглядность и демонстрационный опыт при устном ответе).
7. Использование дополнительного материала (приветствуется, но не обязательно для всех студентов).

²Методические рекомендации для студентов [Электронный ресурс]: Режим доступа: http://lesgaft.spb.ru/sites/default/files/u57/metod.rekomendacii_dlya_studentov_21.pdf

8. Рациональность использования времени, отведенного на задание (не одобряется затянутость выполнения задания, устного ответа во времени, с учетом индивидуальных особенностей студентов)³.

Ответ на каждый вопрос из плана семинарского занятия должен быть содержательным и аргументированным. Для этого следует использовать документы, монографическую, учебную и справочную литературу.

Для успешной подготовки к устному опросу, студент должен законспектировать рекомендуемую литературу, внимательно осмыслить лекционный материал и сделать выводы. В среднем, подготовка к устному опросу по одному семинарскому занятию занимает от 2 до 4 часов в зависимости от сложности темы и особенностей организации обучающимся своей самостоятельной работы.

3. Методические рекомендации по подготовке к практическим занятиям

На практических занятиях необходимо стремиться к самостоятельному решению задач, находя для этого более эффективные методы. При этом студентам надо приучить себя доводить решения задач до конечного «идеального» ответа. Это очень важно для

³Методические рекомендации для студентов [Электронный ресурс]:
http://priab.ru/images/metod_agro/Metod_Inostran_yazyk_35.03.04_Agro_15.01.2016.pdf

будущих специалистов. Практические занятия вырабатывают навыки самостоятельной творческой работы, развивают мыслительные способности.

Практическое занятие – активная форма учебного процесса, дополняющая теоретический курс или лекционную часть учебной дисциплины и призванная помочь обучающимся освоиться в «пространстве» (тематике) дисциплины, самостоятельно прооперировать теоретическими знаниями на конкретном учебном материале.

Продолжительность одного практического занятия – от 2 до 4 академических часов. Общая доля практических занятий в учебном времени на дисциплину – от 10 до 20 процентов (при условии, что все активные формы займут в учебном времени на дисциплину от 40 до 60 процентов).

Для практического занятия в качестве темы выбирается обычно такая учебная задача, которая предполагает не существенные эвристические и аналитические напряжения и продвижения, а потребность обучающегося «потрогать» материал, опознать в конкретном то общее, о чем говорилось в лекции. Например, при рассмотрении вопросов оплаты труда, мотивации труда и проблем безработицы в России имеет смысл провести практические занятия со следующими сюжетами заданий: «Расчет заработной платы работников предприятия». «Разработка механизма мотивации труда на предприятии N». «В чем причины и особенности безработицы в России?». Последняя тема предполагает уже некоторую аналитическую составляющую. Основная задача первой из этих тем - самим посчитать заработную плату для различных групп работников на примере заданных параметров для конкретного предприятия, т. е. сделать расчеты «как на практике»; второй – дать собственный вариант мотивационной политики для предприятия, учитывая особенности данного объекта, отрасли и т.д.; третьей – опираясь на теоретические знания в области проблем занятости и безработицы, а также статистические материалы, сделать авторские выводы о видах безработицы, характерных для России, и их причинах, а также предложить меры по минимизации безработицы.

Перед проведением занятия должен быть подготовлен специальный материал – тот объект, которым обучающиеся станут оперировать, активизируя свои теоретические (общие) знания и тем самым, приобретая навыки выработки уверенных суждений и осуществления конкретных действий.

Дополнительный материал для практического занятия лучше получить у преподавателя заранее, чтобы у студентов была возможность просмотреть его и подготовить вопросы.

Условия должны быть такими, чтобы каждый мог работать самостоятельно от начала до конца. В аудитории должны быть «под рукой» необходимые справочники и тексты законов и нормативных актов по тематике занятия. Чтобы сделать практическое занятие максимально эффективным, надо заранее подготовить и изучить материал по наиболее интересным и практически важным темам.

Особенности практического занятия с использованием компьютера

Для того чтобы повысить эффективность проведения практического занятия, может использоваться компьютер по следующим направлениям:

- поиск информации в Интернете по поставленной проблеме: в этом случае преподаватель представляет обучающимся перечень рекомендуемых для посещения Интернет-сайтов;
- использование прикладных обучающих программ;
- выполнение заданий с использованием обучающимися заранее установленных преподавателем программ;
- использование программного обеспечения при проведении занятий, связанных с моделированием социально-экономических процессов.

4. Методические рекомендации по подготовке семинарским занятиям

Семинар представляет собой комплексную форму и завершающее звено в изучении определенных тем, предусмотренных программой учебной дисциплины. Комплексность данной формы занятий определяется тем, что в ходе её проведения сочетаются выступления обучающихся и преподавателя: рассмотрение обсуждаемой проблемы и анализ различных, часто дискуссионных позиций; обсуждение мнений обучающихся и разъяснение (консультация) преподавателя; углубленное изучение теории и приобретение навыков умения ее использовать в практической работе.

По своему назначению семинар, в процессе которого обсуждается та или иная научная проблема, способствует:

- углубленному изучению определенного раздела учебной дисциплины, закреплению знаний;
- отработке методологии и методических приемов познания;
- выработке аналитических способностей, умения обобщения и формулирования выводов;
- приобретению навыков использования научных знаний в практической деятельности;
- выработке умения кратко, аргументированно и ясно излагать обсуждаемые вопросы;
- осуществлению контроля преподавателя за ходом обучения.

Семинары представляет собой дискуссию в пределах обсуждаемой темы (проблемы). Дискуссия помогает участникам семинара приобрести более совершенные знания, проникнуть в суть изучаемых проблем. Выработать методологию, овладеть методами анализа социально-экономических процессов. Обсуждение должно носить творческий характер с четкой и убедительной аргументацией.

По своей структуре семинар начинается со вступительного слова преподавателя, в котором кратко излагаются место и значение обсуждаемой темы (проблемы) в данной дисциплине, напоминаются порядок и направления ее обсуждения. Конкретизируется ранее известный обучающимся план проведения занятия. После этого начинается процесс обсуждения вопросов обучающимися. Завершается занятие заключительным словом преподавателя.

Проведение семинарских занятий в рамках учебной группы (20 - 25 человек) позволяет обеспечить активное участие в обсуждении проблемы всех присутствующих.

По ходу обсуждения темы помните, что изучение теории должно быть связано с определением (выработкой) средств, путей применения теоретических положений в практической деятельности, например, при выполнении функций государственного служащего. В то же время важно не свести обсуждение научной проблемы только к пересказу случаев из практики работы, к критике имеющих место недостатков. Дискуссии имеют важное значение: учат дисциплине ума, умению выступать по существу, мыслить логически, выделяя главное, критически оценивать выступления участников семинара.

В процессе проведения семинара обучающиеся могут использовать разнообразные по своей форме и характеру пособия (от доски смелом до самых современных технических средств), демонстрируя фактический, в том числе статистический материал, убедительно подтверждающий теоретические выводы и положения. В завершение обсудите результаты работы семинара и сделайте выводы, что хорошо усвоено, а над чем следует дополнительно поработать.

В целях эффективности семинарских занятий необходима обстоятельная подготовка к их проведению. В начале семестра (учебного года) возьмите в библиотеке необходимые методические материалы для своевременной подготовки к семинарам. Во время лекций, связанных с темой семинарского занятия, следует обращать внимание на то, что необходимо дополнительно изучить при подготовке к семинару (новые официальные документы, статьи в периодических журналах, вновь вышедшие монографии и т.д.).

5. Методические рекомендации по подготовке к сдаче экзаменов и зачетов

Экзамен - одна из важнейших частей учебного процесса, имеющая огромное значение.

Во-первых, готовясь к экзамену, студент приводит в систему знания, полученные на лекциях, семинарах, практических и лабораторных занятиях, разбирается в том, что осталось непонятным, и тогда изучаемая им дисциплина может быть воспринята в полном объеме с присущей ей строгостью и логичностью, ее практической направленностью. А это чрезвычайно важно для будущего специалиста.

Во-вторых, каждый хочет быть волевым и сообразительным, выдержанным и целеустремленным, иметь хорошую память, научиться быстро находить наиболее рациональное решение в трудных ситуациях. Очевидно, что все эти качества не только украшают человека, но и делают его наиболее действенным членом коллектива. Подготовка и сдача экзамена помогают студенту глубже усвоить изучаемые дисциплины, приобрести навыки и качества, необходимые хорошему специалисту.

Конечно, успех на экзамене во многом обусловлен тем, насколько систематически и глубоко работал студент в течение семестра. Совершенно очевидно, что серьезно продумать и усвоить содержание изучаемых дисциплин за несколько дней подготовки к экзамену просто невозможно даже для очень способного студента. И, кроме того, хорошо известно, что быстро выученные на память разделы учебной дисциплины так же быстро забываются после сдачи экзамена.

При подготовке к экзамену студенты не только повторяют и дорабатывают материал дисциплины, которую они изучали в течение семестра, они обобщают полученные знания, осмысливают методологию предмета, его систему, выделяют в нем основное и главное, воспроизводят общую картину с тем, чтобы яснее понять связь между отдельными элементами дисциплины. Вся эта обобщающая работа проходит в условиях напряжения воли и сознания, при значительном отвлечении от повседневной жизни, т. е. в условиях благоприятствующих пониманию и запоминанию.

Подготовка к экзаменам состоит в приведении в порядок своих знаний. Даже самые способные студенты не в состоянии в короткий период зачетно-экзаменационной сессии усвоить материал целого семестра, если они над ним не работали в свое время. Для тех, кто мало занимался в семестре, экзамены принесут мало пользы: что быстро пройдено, то быстро и забудется. И хотя в некоторых случаях студент может «проскочить» через экзаменационный барьер, в его подготовке останется серьезный пробел, трудно восполняемый впоследствии.

Определив назначение и роль экзаменов в процессе обучения, попытаемся на этой основе пояснить, как лучше готовиться к ним.

Экзаменам, как правило, предшествует защита курсовых работ (проектов) и сдача зачетов. К экзаменам допускаются только студенты, защитившие все курсовые работы (проекты) и сдавшие все зачеты. В вузе сдача зачетов организована так, что при систематической работе в течение семестра, своевременной и успешной сдаче всех текущих работ, предусмотренных графиком учебного процесса, большая часть зачетов не вызывает повышенной трудности у студента. Студенты, работавшие в семестре по плану, подходят к экзаменационной сессии без напряжения, без излишней затраты сил в последнюю, «зачетную» неделю.

Подготовку к экзамену следует начинать с первого дня изучения дисциплины. Как правило, на лекциях подчеркиваются наиболее важные и трудные вопросы или разделы дисциплины, требующие внимательного изучения и обдумывания. Нужно эти вопросы выделить и обязательно постараться разобраться в них, не дожидаясь экзамена, проработать их, готовясь к семинарам, практическим или лабораторным занятиям, попробовать самостоятельно решить несколько типовых задач. И если, несмотря на это, часть материала осталась неувоенной, ни в коем случае нельзя успокаиваться, надеясь на

то, что это не попадет на экзамене. Факты говорят об обратном; если те или другие вопросы учебной дисциплины не вошли в экзаменационный билет, преподаватель может их задать (и часто задает) в виде дополнительных вопросов.

Точно такое же отношение должно быть выработано к вопросам и задачам, перечисленным в программе учебной дисциплины, выдаваемой студентам в начале семестра. Обычно эти же вопросы и аналогичные задачи содержатся в экзаменационных билетах. Не следует оставлять без внимания ни одного раздела дисциплины: если не удалось в чем-то разобраться самому, нужно обратиться к товарищам; если и это не помогло выяснить какой-либо вопрос до конца, нужно обязательно задать этот вопрос преподавателю на предэкзаменационной консультации. Чрезвычайно важно приучить себя к умению самостоятельно мыслить, учиться думать, понимать суть дела. Очень полезно после проработки каждого раздела восстановить в памяти содержание изученного материала, кратко записав это на листе бумаги, создать карту памяти (умственную карту), изобразить необходимые схемы и чертежи (логико-графические схемы), например, отобразить последовательность вывода теоремы или формулы. Если этого не сделать, то большая часть материала останется не понятой, а лишь формально заученной, и при первом же вопросе экзаменатора студент убедится в том, насколько поверхностно он усвоил материал.

В период экзаменационной сессии происходит резкое изменение режима работы, отсутствует посещение занятий по расписанию. При всяком изменении режима работы очень важно скорее приспособиться к новым условиям. Поэтому нужно сразу выбрать такой режим работы, который сохранился бы в течение всей сессии, т. е. почти на месяц. Необходимо составить для себя новый распорядок дня, чередуя занятия с отдыхом. Для того чтобы сократить потерю времени на включение в работу, рабочие периоды целесообразно делать длительными, разделив день примерно на три части: с утра до обеда, с обеда до ужина и от ужина до сна.

Каждый рабочий период дня надо заканчивать отдыхом. Наилучший отдых в период экзаменационной сессии - прогулка, кратковременная пробежка или какой-либо неустойчивый физический труд.

При подготовке к экзаменам основное направление дают программа учебной дисциплины и студенческий конспект, которые указывают, что наиболее важно знать и уметь делать. Основной материал должен прорабатываться по учебнику (если такой имеется) и учебным пособиям, так как конспекта далеко недостаточно для изучения дисциплины. Учебник должен быть изучен в течение семестра, а перед экзаменом сосредоточьте внимание на основных, наиболее сложных разделах. Подготовку по каждому разделу следует заканчивать восстановлением по памяти его краткого содержания в логической последовательности.

За один - два дня до экзамена назначается консультация. Если ее правильно использовать, она принесет большую пользу. Во время консультации студент имеет полную возможность получить ответ на нее и ясные ему вопросы. А для этого он должен проработать до консультации все темы дисциплины. Кроме того, преподаватель будет отвечать на вопросы других студентов, что будет для вас повторением и закреплением знаний. И еще очень важное обстоятельство: преподаватель на консультации, как правило, обращает внимание на те вопросы, по которым на предыдущих экзаменах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных темах дисциплины. Некоторые студенты не приходят на консультации либо потому, что считают, что у них нет вопросов к преподавателю, либо полагают, что у них и так мало времени и лучше самому прочитать материал в конспекте или в учебнике. Это глубокое заблуждение. Никакая другая работа не сможет принести столь значительного эффекта накануне экзамена, как консультация преподавателя.

Но консультация не может возместить отсутствия длительной работы в течение семестра и помочь за несколько часов освоить материал, требующийся к экзамену. На консультации студент получает ответы на трудные или оставшиеся неясными вопросы и, следовательно, дорабатывается материал. Консультации рекомендуется посещать,

подготовив к ним все вопросы, вызывающие сомнения. Если студент придет на консультацию, не проработав всего материала, польза от такой консультации будет невелика.

Очень важным условием для правильного режима работы в период экзаменационной сессии является нормальный сон. Подготовка к экзамену не должна идти в ущерб сну, иначе в день экзамена не будет чувства свежести и бодрости, необходимых для хороших ответов. Вечер накануне экзамена рекомендуем закончить небольшой прогулкой.

Итак, *основные советы* для подготовки к сдаче зачетов и экзаменов состоят в следующем:

- лучшая подготовка к зачетам и экзаменам - равномерная работа в течение всего семестра;
- используйте программы учебных дисциплин - это организует вашу подготовку к зачетам и экзаменам;
- учитывайте, что для полноценного изучения учебной дисциплины необходимо время;
- составляйте планы работы во времени;
- работайте равномерно и ритмично;
- курсовые работы (проекты) желательно защищать за одну - две недели до начала зачетно-экзаменационной сессии;
- все зачеты необходимо сдавать до начала экзаменационной сессии;
- помните, что конспект не заменяет учебник и учебные пособия, а помогает выбрать из него основные вопросы и ответы;
- при подготовке наибольшее внимание и время уделяйте трудным и непонятным вопросам учебной дисциплины;
- грамотно используйте консультации;
- соблюдайте правильный режим труда и отдыха во время сессии, это сохранит работоспособность и даст хорошие результаты;
- учитесь владеть собой на зачете и экзамене;
- учитесь точно и кратко передавать свои мысли, поясняя их, если нужно, логико-графическими схемами.

ЗАКЛЮЧЕНИЕ

Методические указания по выполнению самостоятельной работы обучающихся являются неотъемлемой частью процесса обучения в вузе. Правильная организация самостоятельной работы позволяет обучающимся развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, способствует формированию навыков совершенствования профессионального мастерства. Также внеаудиторное время включает в себя подготовку к аудиторным занятиям и изучение отдельных тем, расширяющих и углубляющих представления обучающихся по разделам изучаемой дисциплины.

Таким образом, обучающийся используя методические указания может в достаточном объеме усвоить и успешно реализовать конкретные знания, умения, навыки и получить опыт при выполнении следующих условий:

- 1) систематическая самостоятельная работа по закреплению полученных знаний и навыков;
- 2) добросовестное выполнение заданий;
- 3) выяснение и уточнение отдельных предпосылок, умозаключений и выводов, содержащихся в учебном курсе;
- 4) сопоставление точек зрения различных авторов по затрагиваемым в учебном курсе проблемам; выявление неточностей и некорректного изложения материала в периодической и специальной литературе;
- 5) периодическое ознакомление с последними теоретическими и практическими достижениями в области управления персоналом;
- 6) проведение собственных научных и практических исследований по одной или нескольким актуальным проблемам для *HR*;
- 7) подготовка научных статей для опубликования в периодической печати, выступление на научно-практических конференциях, участие в работе студенческих научных обществ, круглых столах и диспутах по проблемам управления персоналом.

СПИСОК ЛИТЕРАТУРЫ

1. Брандес М. П. Немецкий язык. Переводческое реферирование: практикум. М.: КДУ, 2008. – 368с.
2. Долгоруков А. Метод case-study как современная технология профессионально-ориентированного обучения [Электронный ресурс]. Режим доступа: <http://evolkov.net/case/case.study.html>
3. Методические рекомендации по написанию реферата. [Электронный ресурс]. Режим доступа: <http://www.hse.spb.ru/edu/recommendations/method-referat-2005.phtml>
4. Фролова Н. А. Реферирование и аннотирование текстов по специальности: Учеб. пособие / ВолгГТУ, Волгоград, 2006. - С.5.
5. Методические рекомендации для студентов [Электронный ресурс]: Режим доступа: http://lesgaft.spb.ru/sites/default/files/u57/metod.rekomendacii_dlya_studentov_21.pdf

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «Уральский государственный горный университет»



УТВЕРЖДАЮ

И.о. проректора по учебно-методической работе _____ В.В. Зубов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ И ЗАДАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

ФТД.В.04 ОСНОВЫ ВОЕННОЙ ПОДГОТОВКИ

Специальность

21.05.04 Горное дело

Специализация

Мехатроника и робототехника промышленных комплексов

Одобрены на заседании кафедры

Геологии и защиты в чрезвычайных ситуациях

(название кафедры)

Зав. кафедрой _____

(подпись)

Стороженко Л.А.

(Фамилия И.О.)

Протокол № 1 от 11.09.2024

(Дата)

Рассмотрены методической комиссией
факультета

Горно-механического

(название факультета)

Председатель _____

(подпись)

Осипов П. А.

(Фамилия И.О.)

Протокол № 2 от 18.10.2024

(Дата)

Екатеринбург

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	5
2. СОДЕРЖАНИЕ РАЗДЕЛОВ (ТЕМ) ДИСЦИПЛИНЫ	5
3. ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ОПРОСУ.....	9
4. ТЕСТОВЫЕ ЗАДАНИЯ.....	10
5. ПОДГОТОВКА К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	28
6. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	31

ВВЕДЕНИЕ

Самостоятельная работа в высшем учебном заведении - это часть учебного процесса, метод обучения, прием учебно-познавательной деятельности, комплексная целевая стандартизованная учебная деятельность с запланированными видом, типом, формами контроля.

Самостоятельная работа представляет собой плановую деятельность обучающихся по поручению и под методическим руководством преподавателя.

Целью самостоятельной работы студентов является закрепление тех знаний, которые они получили на аудиторных занятиях, а также способствование развитию у студентов творческих навыков, инициативы, умению организовать свое время.

Самостоятельная работа реализует следующие задачи:

- предполагает освоение курса дисциплины;
- помогает освоению навыков учебной работы;
- способствует осознанию ответственности процесса познания;
- способствует углублению и пополнению знаний студентов, освоению ими навыков и умений;
- формирует интерес к познавательным действиям, освоению методов и приемов познавательного процесса,
- создает условия для творческой и научной деятельности обучающихся;
- способствует развитию у студентов таких личных качеств, как целеустремленность, заинтересованность, исследование нового.

Самостоятельная работа обучающегося выполняет следующие функции:

- развивающую (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей студентов);
- информационно-обучающую (учебная деятельность студентов на аудиторных занятиях, неподкрепленная самостоятельной работой, становится мало результативной);
- ориентирующую и стимулирующую (процессу обучения придается ускорение и мотивация);
- воспитательную (формируются и развиваются профессиональные качества специалиста и гражданина);
- исследовательскую (новый уровень профессионально-творческого мышления).

Организация самостоятельной работы студентов должна опираться на определенные требования, а, именно:

- сложность осваиваемых знаний должна соответствовать уровню развития студентов;
- стандартизация заданий в соответствии с логической системой курса дисциплины;
- объем задания должен соответствовать уровню студента;
- задания должны быть адаптированными к уровню студентов.

Содержание самостоятельной работы студентов представляет собой, с одной стороны, совокупность теоретических и практических учебных заданий, которые должен выполнить студент в процессе обучения, объект его деятельности; с другой стороны - это способ деятельности студента по выполнению соответствующего теоретического или практического учебного задания.

Свое внешнее выражение содержание самостоятельной работы студентов находит во всех организационных формах аудиторной и внеаудиторной деятельности, в ходе самостоятельного выполнения различных заданий.

Функциональное предназначение самостоятельной работы студентов в процессе лекций, практических занятий по овладению специальными знаниями заключается в самостоятельном прочтении, просмотре, прослушивании, наблюдении, конспектировании, осмыслении, запоминании и воспроизведении определенной информации. Цель и планирование самостоятельной работы студента определяет преподаватель. Вся информация осуществляется на основе ее воспроизведения.

Так как самостоятельная работа тесно связана с учебным процессом, ее необходимо рассматривать в двух аспектах:

1. аудиторная самостоятельная работа – практические занятия;
2. внеаудиторная самостоятельная работа – подготовка к практическим занятиям.

Основные формы организации самостоятельной работы студентов определяются следующими параметрами:

- содержание учебной дисциплины;
- уровень образования и степень подготовленности студентов;
- необходимость упорядочения нагрузки студентов при самостоятельной работе.

Таким образом, самостоятельная работа студентов является важнейшей составной частью процесса обучения.

Методические указания по организации самостоятельной работы и задания для обучающихся по дисциплине «*Основы военной подготовки*» обращают внимание студента на главное, существенное в изучаемой дисциплине, помогают выработать умение анализировать явления и факты, связывать теоретические положения с практикой, а также облегчают подготовку к сдаче *зачета*.

Настоящие методические указания позволяют студентам самостоятельно овладеть фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю подготовки, опытом творческой и исследовательской деятельности, и направлены на формирование компетенций, предусмотренных учебным планом поданному профилю.

Видами самостоятельной работы обучающихся по дисциплине «*Основы военной подготовки*» являются:

- самостоятельное изучение тем курса (в т.ч. рассмотрение основных категорий дисциплины, работа с литературой);
- подготовка к практическим занятиям (в т.ч. ответы на вопросы для самопроверки, подготовка к выполнению практических заданий);

- подготовка к зачету.

В методических указаниях представлены материалы для самостоятельной работы и рекомендации по организации отдельных её видов.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В современных условиях подготовка граждан Российской Федерации к военной службе является приоритетным направлением государственной политики. Важнейшими вопросами образования на всех уровнях является воспитание любви к Родине, чувства патриотизма, готовности к защите Отечества.

Образовательная дисциплина «Основы военной подготовки» (далее – дисциплина) реализуется исходя из базовых принципов и направлений военной подготовки, дисциплина состоит из основных разделов военной подготовки, тем военно-политической и правовой подготовки.

Основной целью освоения дисциплины является получение знаний, умений и навыков, необходимых для становления обучающихся образовательных организаций высшего образования (далее – вуз) в качестве граждан способных и готовых к выполнению воинского долга и обязанности по защите своей Родины в соответствии с законодательством Российской Федерации.

Задачами дисциплины «Основы военной подготовки» являются:

- 1) формирование у обучающихся понимания главных положений военной доктрины Российской Федерации, а также основ военного строительства и структуры Вооруженных Сил Российской Федерации (далее - ВС РФ);
- 2) формирование у обучающихся высокого общественного сознания и воинского долга;
- 3) воспитание дисциплинированности, высоких морально-психологических качеств личности гражданина – патриота;
- 4) освоение базовых знаний и формирование ключевых навыков военного дела;
- 5) раскрытие специфики деятельности различных категорий военнослужащих ВС РФ;
- 6) ознакомление с нормативными документами в области обеспечения обороны государства и прохождения военной службы;
- 7) формирование строевой подтянутости, уважительного отношения к воинским ритуалам и традициям, военной форме одежды;
- 8) изучение и принятие правил воинской вежливости;
- 9) овладение знаниями уставных норм и правил поведения военнослужащих.

2. СОДЕРЖАНИЕ РАЗДЕЛОВ (ТЕМ) ДИСЦИПЛИНЫ

Раздел 1. Общевоинские уставы Вооруженных Сил Российской Федерации

Тема 1. Общевоинские уставы Вооруженных Сил Российской Федерации, их основные требования и содержание.

Структура, требования и основное содержание общевоинских уставов. Права военнослужащих. Общие обязанности военнослужащих. Воинские звания. Единоначалие. Начальники и подчиненные. Старшие и младшие. Приказ и приказание. Порядок отдачи и выполнение приказа. Воинская вежливость и воинская дисциплина военнослужащих.

Тема 2. Внутренний порядок и суточный наряд.

Размещение военнослужащих. Распределение времени и внутренний порядок. Суточный наряд роты, его предназначение, состав.

Дневальный, дежурный по роте. Развод суточного наряда.

Тема 3. Общие положения Устава гарнизонной и караульной службы.

Общие положения Устава гарнизонной и караульной службы.

Обязанности разводящего, часового.

Раздел 2. Строевая подготовка

Тема 4. Строевые приемы и движение без оружия.

Строй и его элементы. Виды строя. Сигналы для управления строем. Команды и порядок их подачи. Обязанности командиров, военнослужащих перед построением и в строю. Строевой расчет. Строевая стойка. Выполнение команд: «Становись», «Равняйсь», «Смирно», «Вольно», «Заправиться». Повороты на месте. Строевой шаг. Движение строевым шагом. Движение строевым шагом в составе подразделения. Повороты в движении. Движение в составе взвода. Управление подразделением в движении.

Раздел 3. Огневая подготовка из стрелкового оружия

Тема 5. Основы, приемы и правила стрельбы из стрелкового оружия.

Требования безопасности при обращении со стрелковым оружием.

Требования безопасности при проведении занятий по огневой подготовке.

Приемы и правила стрельбы из стрелкового оружия.

Тема 6. Назначение, боевые свойства, материальная часть и применение стрелкового оружия, ручных противотанковых гранатометов и ручных гранат.

Назначение, состав, боевые свойства и порядок сборки разборки АК-74 и РПК-74.

Назначение, состав, боевые свойства и порядок сборки разборки пистолета ПМ.

Назначение, состав, боевые свойства РПГ-7.

Назначение, боевые свойства и материальная часть ручных гранат.

Сборка разборка пистолета ПМ и подготовка его к боевому применению.

Сборка разборка АК-74, РПК-74 и подготовка их к боевому применению.

Снаряжение магазинов и подготовка ручных гранат к боевому применению.

Тема 7. Выполнение упражнений учебных стрельб из стрелкового оружия. Требования безопасности при организации и проведении стрельб из стрелкового оружия. Порядок выполнения упражнения учебных стрельб. Меры безопасности при проведении стрельб и проверка усвоения знаний и мер безопасности при обращении со стрелковым оружием. Выполнение норматива №1 курса стрельб из стрелкового оружия.

Раздел 4. Основы тактики общевойсковых подразделений

Тема 8. Вооруженные Силы Российской Федерации их состав и задачи. Тактико-технические характеристики (ТТХ) основных образцов вооружения и техники ВС РФ. Вооруженные Силы Российской Федерации их состав и задачи. Назначение, структура мотострелковых и танковых подразделений сухопутных войск, их задачи в бою. Боевое предназначение входящих в них подразделений. Тактико-технические характеристики основных образцов вооружения и техники ВС РФ.

Тема 9. Основы общевойскового боя.

Сущность современного общевойскового боя, его характеристики и виды. Способы ведения современного общевойскового боя и средства вооруженной борьбы.

Тема 10. Основы инженерного обеспечения.

Цели и основные задачи инженерного обеспечения частей и подразделений. Назначение, классификация инженерных боеприпасов, инженерных заграждений и их характеристики.

Полевые фортификационные сооружения: окоп, траншея, ход сообщения, укрытия, убежища.

Тема 11. Организация воинских частей и подразделений, вооружение, боевая техника вероятного противника.

Организация, вооружение, боевая техника подразделений мпб и тб армии США.

Организация, вооружение, боевая техника подразделений мпб и тб армии Германии.

Раздел 5. Радиационная, химическая и биологическая защита

Тема 12. Ядерное, химическое, биологическое, зажигательное оружие.

Ядерное оружие. Средства их применения. Поражающие факторы ядерного взрыва и их воздействие на организм человека, вооружение, технику и фортификационные сооружения.

Химическое оружие. Отравляющие вещества (ОВ), их назначение, классификация и воздействие на организм человека. Боевые состояния, средства применения, признаки применения ОВ, их стойкость на местности.

Биологическое оружие. Основные виды и поражающее действие.

Средства применения, внешние признаки применения.

Зажигательное оружие. Поражающие действия зажигательного оружия на личный состав, вооружение и военную технику, средства и способы защиты от него.

Тема 13. Радиационная, химическая и биологическая защита.

Цель, задачи и мероприятия РХБ защиты.

Мероприятия специальной обработки:

дегазация, дезактивация, дезинфекция, санитарная обработка.

Цели и порядок проведения частичной и полной специальной обработки.

Технические средства и приборы радиационной, химической и биологической защиты. Средства индивидуальной защиты и порядок их использования.

Подгонка и техническая проверка средств индивидуальной защиты.

Раздел 6. Военная топография

Тема 14. Местность как элемент боевой обстановки.

Измерения и ориентирование на местности без карты, движение по азимутам.

Местность как элемент боевой обстановки.

Способы ориентирования на местности без карты.

Способы измерения расстояний. Движение по азимутам.

Тема 15. Топографические карты и их чтение, подготовка к работе.

Определение координат объектов и целеуказания по карте.

Геометрическая сущность, классификация и назначение топографических карт.

Определение географических и прямоугольных координат объектов по карте.

Целеуказание по карте.

Раздел 7. Основы медицинского обеспечения

Тема 16. Медицинское обеспечение войск (сил), первая медицинская помощь при ранениях, травмах и особых случаях.

Медицинское обеспечение – как вид всестороннего обеспечения войск.

Обязанности и оснащение должностных лиц медицинской службы тактического звена в бою. Общие правила оказания самопомощи и взаимопомощи.

Первая помощь при ранениях и травмах.

Первая помощь при поражении отравляющими веществами, бактериологическими средствами. Содержание мероприятия доврачебной помощи.

Раздел 8. Военно-политическая подготовка

Тема 17. Россия в современном мире. Основные направления социально-экономического, политического и военно-технического развития страны.

Новые тенденции и особенности развития современных международных отношений.

Место и роль России в многополярном мире.

Основные направления социально-экономического, политического и военно-технического развития Российской Федерации.

Цели, задачи, направления и формы военно-политической работы в подразделении, требования руководящих документов.

Раздел 9. Правовая подготовка

Тема 18. Военная доктрина Российской Федерации. Законодательство Российской Федерации о прохождении военной службы.

Основные положения Военной доктрины Российской Федерации.

Правовая основа воинской обязанности и военной службы.

Понятие военной службы, ее виды и их характеристики.

Обязанности граждан по воинскому учету.

3. ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ОПРОСУ

Раздел 3. Огневая подготовка из стрелкового оружия

1. Каковы виды стрелкового оружия
2. Какие бывают боеприпасы
3. Марки ручных гранат.

Раздел 4. Основы тактики общевойсковых подразделений

1. Из чего состоит организационно-штатная структура общевойсковых подразделений
2. Перечислите Тактико-технические характеристики (ТТХ) основных образцов вооружения и техники ВС РФ
3. Каковы основные факторы, определяющие характер, организацию и способы ведения современного общевойскового боя
4. Перечислите основные инженерно-технических мероприятия.
5. Какие знаете защитные сооружения.
6. Виды заграждений
7. На чем основывается полевое водоснабжение.
8. Каковы емкости РДВ
9. Назначение ТУФ-200
10. Назначение МТК
11. Перечислите ТТХ и ТТД вооружения и боевой техники армии США
12. Перечислите ТТХ и ТТД вооружения и боевой техники армии Германии

Раздел 5. Радиационная, химическая и биологическая защита

1. Общие сведения о ядерном оружии
2. Общие сведения о химическом оружии
3. Общие сведения о биологическом оружии
4. Каковы правила поведения и меры профилактики в условиях радиоактивного заражения
5. Каковы правила поведения и меры профилактики при применении отравляющих веществ

6. Каковы правила поведения и меры профилактики в условиях применения бактериальных средств
7. Какие знаете индивидуальные средства РХБ защиты
8. Каковы мероприятия радиационной, химической и биологической защиты?

Раздел 6. Военная топография

1. Каковы тактические свойства местности, их влияние на действия подразделений в боевой обстановке
2. Назначение, номенклатура и условные знаки топографических карт
3. Способы ориентирования на местности по карте и без карты
4. Что такое номенклатура топографических карт
5. Как задается координаты объекта
6. Уточнение координат по "улитке"

Раздел 8. Военно-политическая подготовка

1. Каковы тенденции и особенности развития современных международных отношений, место и роль России в многополярном мире, основные направления социально-экономического, политического и военно-технического развития страны, основные положения Военной доктрины РФ
2. Правовое положение и порядок прохождения военной службы.

Раздел 9. Правовая подготовка

1. Что значит нормативно-правовой акт
2. Чем определяется порядок прохождения военной службы

4. ТЕСТОВЫЕ ЗАДАНИЯ

1. Кем утверждаются ОВУ?

- **Вариант ответа**

Государственной Думой

- **Вариант ответа**

Правительством РФ

- **Вариант ответа**

Федеральным собранием РФ

- **Вариант ответа**

Президентом РФ

2. Кем присваивается первое офицерское звание?

- **Вариант ответа**

Мин. Обороны РФ

- **Вариант ответа**

Президентом РФ

- **Вариант ответа**

Правительством РФ

- **Вариант ответа**

Командующим ВВО

- **Вариант ответа**

Командиром в/ч

3. К какому виду ответственности могут быть привлечены офицеры запаса за уклонение от призыва на действительную в/службу?

- **Вариант ответа**

моральной

- **Вариант ответа**

нравственной

- **Вариант ответа**

административной

- **Вариант ответа**

уголовной

- **Вариант ответа**

общественной

4. На чем основываются взаимоотношения между военнослужащими?

- **Вариант ответа**

на дружбе

- **Вариант ответа**

на чувстве взаимного уважения

- **Вариант ответа**

на товариществе

- **Вариант ответа**

на любви

- **Вариант ответа**

на вере в силу армии РФ

5. Какое минимальное количество часов отдыха разрешено военнослужащим согласно распорядку дня?

- **Вариант ответа**

не > 10

- **Вариант ответа**

не < 8

- **Вариант ответа**

не > 8

- **Вариант ответа**

не > 6

- **Вариант ответа**

не < 6

6. Какое минимальное количество унитазов (очек) должно быть в подразделении (поте)?

- **Вариант ответа**

не < 15 шт

- Вариант ответа

1 на 30 чел

- Вариант ответа

не > 20 шт

- Вариант ответа

1 на 20 чел

- Вариант ответа

1 на 10-12 чел

7. Сколько должно быть умывальников для умывания в подразделении (роте)?

- Вариант ответа

1 на 5-7 чел

- Вариант ответа

не > 10 шт

- Вариант ответа

не < 10 шт

- Вариант ответа

1 на 10 чел

- Вариант ответа

1 на 8-9 чел

8. Сколько форм одежды для проведения утренней физзарядки?

- Вариант ответа

1

- Вариант ответа

2

- Вариант ответа

3

- Вариант ответа

6

- Вариант ответа

5

9. На чем основывается воинская дисциплина?

- Вариант ответа

на чувстве ответственности за порученное дело

- Вариант ответа

на страхе перед командиром

- Вариант ответа

на осознании каждым в/с воинского долга и личной ответственности за защиту своего Отечества, на его беззаветной преданности своему народу

- Вариант ответа

на взаимном уважении военнослужащих

- **Вариант ответа**
на чувстве преданности Родине

10. Какие поощрения могут применяться к младшим офицерам?

- **Вариант ответа**
внеочередной отпуск до 10 суток

- **Вариант ответа**
увольнение

- **Вариант ответа**
благодарность, награждение грамотой, присвоение внеочередного воинского звания

- **Вариант ответа**
фотографированием у развернутого б/знамени части

- **Вариант ответа**
награждение путевкой в санаторий

11. Какие взыскания могут накладываться на мл. офицеров?

- **Вариант ответа**
до 3 нарядов вне очереди на службу

- **Вариант ответа**
лишение увольнения

- **Вариант ответа**
арест с содержанием на г/вахте

- **Вариант ответа**
отлучение от офицерского собрания

- **Вариант ответа**
выговор, строгий выговор, снятие с должности, понижение в в/звании

12. Каким взысканием может подвергнуться солдат, проходящий военную службу по призыву, за нарушение воинской дисциплины?

- **Вариант ответа**
арест с содержанием на г/вахте до 30 сут

- **Вариант ответа**
арест с содержанием на г/вахте до 20 сут

- **Вариант ответа**
лишение очередного отпуска на родину

- **Вариант ответа**
задержка в выплате денежного довольствия

- **Вариант ответа**
выговор, строгий выговор, лишение очередного увольнения, до 10 суток ареста с содержанием на г/вахте, лишением нагрудного знака отличника, снижением в должности в в/звании на одну ступень, с переводом на низшую должность, до 5 нарядов вне очереди на работу

13. Каким взысканиям может подвергаться КО за нарушения воинской дисциплины?

- **Вариант ответа**

снятие с должности, выговор, строгий выговор

- **Вариант ответа**

лишение в/звания «сержант»

- **Вариант ответа**

лишение очередного отпуска

- **Вариант ответа**

арест на г/вахту до 5 суток

- **Вариант ответа**

лишение денежного довольствия

14. Какие поощрения могут применяться к солдату?

- **Вариант ответа**

до 10 увольнений вне очереди

- **Вариант ответа**

снятие ранее наложенного взыскания, благодарность, награждение грамотой, ценным подарком, присвоение в/звания «ефрейтор», фотографированием у раз-
вернутого б/знамени

- **Вариант ответа**

до 15 суток кратковременного отпуска

- **Вариант ответа**

награждение боевым именованным оружием

- **Вариант ответа**

выдача дополнительного продовольственного пайка

15. Обязанности КВ по отношению к подчиненным при их поощрении?

- **Вариант ответа**

присваивать в/звания «мл. сержант», «сержант», «ст. сержант»

- **Вариант ответа**

отправлять в отпуск на родину на срок до 10 сут

- **Вариант ответа**

объявлять благодарность, снятие ранее наложенного взыскания

- **Вариант ответа**

освободить с г/вахты

- **Вариант ответа**

водить в чайную за свое денежное довольствие

16. Права КВ по отношению к подчиненным при применении дисциплинарных взысканий?

- **Вариант ответа**

арестовывать на срок до 3 суток

- **Вариант ответа**

лишать отпуска с выездом на Родину

- **Вариант ответа**

объявлять до 6 нарядов вне очереди

- **Вариант ответа**

объявлять выговор, стр. выговор, лишать очередного увольнения, объявлять до 4 нарядов вне очереди на работу

- **Вариант ответа**

лишать выдачи печенья и сахара вместо сигарет некурящим в/служащим

17. Чем вооружается наряд по роте?

- **Вариант ответа**

АК

- **Вариант ответа**

РПГ

- **Вариант ответа**

лопатами

- **Вариант ответа**

вениками

- **Вариант ответа**

штык-ножами

18. Основные задачи наряда по КПП

- **Вариант ответа**

открывать ворота КПП

- **Вариант ответа**

открывать и закрывать ворота КПП

- **Вариант ответа**

смотреть на проходящих мимо красивых девушек

- **Вариант ответа**

осуществлять строгий контрольно-пропускной режим в части

- **Вариант ответа**

пропускать только в/служащих и членов семей на территорию части

19. Основная задача наряда по роте

- **Вариант ответа**

никуда не сбегать

- **Вариант ответа**

мыть полы в спальнях помещений

- **Вариант ответа**

следить за соблюдением распорядка дня, сохранностью КДХО, имущества роты и личных вещей в/служащих

- **Вариант ответа**

находиться в роте

- **Вариант ответа**

убирать грязную посуду за л/с роты в столовой

20. Какие бывают парки в ВС РФ?

- **Вариант ответа**

зоопарки

- **Вариант ответа**

развлечений и отдыха

- **Вариант ответа**

постоянные и полевые

- **Вариант ответа**

аквапарки

- **Вариант ответа**

стационарные

21. Кто назначается в наряд по парку?

- **Вариант ответа**

дежурный и дневальный

- **Вариант ответа**

дежурный и водитель-механик дежурного тягача

- **Вариант ответа**

дежурный, дневальные и барабанщик

- **Вариант ответа**

дежурный, дневальные, механик-водитель дежурного тягача

- **Вариант ответа**

механик-водитель дежурного тягача и барабанщик

22. Состав суточного наряда при перевозках войск

- **Вариант ответа**

дежурный по эшелону, помощник, дежурные по вагонам и дневальные

- **Вариант ответа**

дежурные по вагонам, дневальные, барабанщик, фельдшер

- **Вариант ответа**

дежурный, горнист, фельдшер

- **Вариант ответа**

барабанщик и горнист

- **Вариант ответа**

дневальные

23. Сколько времени должно отводиться распорядком дня подготовки лиц суточного наряда

- **Вариант ответа**

не менее 3 ч

- **Вариант ответа**

не более 1 ч

- **Вариант ответа**
не менее 4 ч
- **Вариант ответа**
не более 1 ч
- **Вариант ответа**
не менее 30 мин

24. С какого момента караул переходит в подчинение дежурному по части

- **Вариант ответа**
после того, как помощник дежурного по части дает команду ”равнение на —...“ и “смирно” и докладывает дежурному по воинской части
- **Вариант ответа**
при выходе дежурного по части на строевой плац
- **Вариант ответа**
когда дежурный по части выдает пароль
- **Вариант ответа**
при докладе о смене караулов
- **Вариант ответа**
при первом прибытии в караул дежурного по части

25. С чего начинается прием дежурства дежурным по роте

- **Вариант ответа**
со счета кроватей в подразделении
- **Вариант ответа**
со счета тумбочек в подразделении
- **Вариант ответа**
с проверки количества оружия и его качественного состояния
- **Вариант ответа**
с перекура
- **Вариант ответа**
с убытием на улицу

26. Кто инструктирует дежурного по парку

- **Вариант ответа**
командир части
- **Вариант ответа**
начальник штаба в/ч
- **Вариант ответа**
начальник штаба батальона
- **Вариант ответа**
командиром батальона
- **Вариант ответа**
заместителем командира части по вооружению

27. Что не должен принимать дежурный по парку

- **Вариант ответа**

объекты, охраняемые внутренним караулом

- **Вариант ответа**

количество деревьев на территории парка

- **Вариант ответа**

количество автомобильной техники в/ч

- **Вариант ответа**

состояние деж. тягача

- **Вариант ответа**

количество и состояние оттисков печатей на хранилищах с техникой

28. Где должно храниться оружие в подразделении

- **Вариант ответа**

под кроватями

- **Вариант ответа**

под подушками

- **Вариант ответа**

за тумбочками

- **Вариант ответа**

в каптерке

- **Вариант ответа**

в комнате для хранения оружия, в пирамидах

29. У кого должны храниться ключи от комнаты для хранения оружия

- **Вариант ответа**

у дежурного по роте

- **Вариант ответа**

у командира взвода

- **Вариант ответа**

у командира батальона

- **Вариант ответа**

у командира роты

- **Вариант ответа**

у дежурного по части

30. Кто дает разрешение на вскрытие комнате для хранения оружия

- **Вариант ответа**

старшина роты

- **Вариант ответа**

командир взвода

- **Вариант ответа**

заместитель командира роты по воспитательной работе

- **Вариант ответа**

командир роты
- **Вариант ответа**
командир части

31. Чьими печатями должна опечатываться комната для хранения оружия

- **Вариант ответа**

командира части и командира роты

- **Вариант ответа**

командира роты и дежурного по роте

- **Вариант ответа**

дежурного по роте и дежурного по части

- **Вариант ответа**

старшины роты и командира роты

- **Вариант ответа**

командира роты и дежурного по части

Тесты по медицинской подготовке:

1. Объем первой медицинской помощи с динамическими (механическими) факторами поражения:

- **Вариант ответа**

временная остановка кровотечения, искусственное дыхание

- **Вариант ответа**

временная остановка наружного кровотечения, устранение асфиксии, искусственное дыхание, непрямой массаж сердца, введение обезболивающих средств, наложение асептических повязок, транспортная иммобилизация

- **Вариант ответа**

простейшие противошоковые мероприятия, временная остановка кровотечения, эвакуация

- **Вариант ответа**

закрытие ран повязками, иммобилизация конечностей табельными и подручными средствами

2. Методы временной остановки наружного кровотечения на месте поражения:

- **Вариант ответа**

наложение асептической повязки, пальцевое прижатие кровоточащего сосуда, давящая повязка, наложение жгута

- **Вариант ответа**

наложение кровоостанавливающего жгута, давящая повязка, тугая тампонада раны, форсированное сгибание конечности с последующей фиксацией, пальцевое прижатие кровоточащего сосуда

- **Вариант ответа**

давящая повязка, наложение жгута, наложение зажима на кровоточащий сосуд, форсированное сгибание конечности

- **Вариант ответа**

наложение зажима на кровоточащий сосуд, наложение асептической повязки

3. Объем первой медицинской помощи при проникающем ранении живота:

- **Вариант ответа**

при эвентрации вправление выпавших органов в брюшную полость, наложение асептической повязки на рану, эвакуация в положении лежа

- **Вариант ответа**

введение анальгетиков, теплое питье, асептическая повязка, эвакуация на носилках

- **Вариант ответа**

обезболивание, наложение асептической повязки, вынос из очага на носилках, эвакуация в первую очередь

- **Вариант ответа**

введение аналептиков, теплое питье, асептическая повязка, эвакуация на носилках

4. Первая медицинская помощь при повреждении таза и тазовых органов:

- **Вариант ответа**

инъекция спазмолитиков, наложение асептических повязок, щадящая эвакуация на носилках на спине, при подозрении на перелом таза - в положении лягушки

- **Вариант ответа**

инъекция промедола, наложение асептических повязок на раны, дача противобактериальных препаратов, эвакуация в положении лягушки

- **Вариант ответа**

инъекция морфина, наложение асептических повязок на раны, при кровотечении прием гомеостатических средств, эвакуация на носилках спиной вниз

- **Вариант ответа**

инъекция морфина, наложение асептических повязок на раны, при кровотечении прием гомеостатических средств, эвакуация на носилках лежа

5. Максимально допустимая длительность клинической смерти:

- **Вариант ответа**

1-2 минуты

- **Вариант ответа**

5-7 минут

- **Вариант ответа**

3-5 минут

- **Вариант ответа**

10-15 минут

6. Порядок реанимационных мероприятий одним человеком:

- **Вариант ответа**

2 вдувания + 30 компрессий

- **Вариант ответа**

1 вдувание + 5 компрессий

- Вариант ответа

3 вдувания + 10 компрессий

- Вариант ответа

5 вдуваний + 20 компрессий

7. Положение рук реаниматора при проведении непрямого массажа сердца:

- Вариант ответа

лучезапястные и локтевые суставы максимально разогнуты

- Вариант ответа

лучезапястные и локтевые суставы максимально согнуты

- Вариант ответа

локтевые суставы согнуты, лучезапястные – разогнуты

- Вариант ответа

локтевые суставы разогнуты, лучезапястные – согнуты

8. Темп непрямого массажа сердца должен составлять в минуту:

- Вариант ответа

5 сжатий

- Вариант ответа

12 сжатий

- Вариант ответа

20 сжатий

- Вариант ответа

100-110 сжатий

9. Реанимация это:

- Вариант ответа

раздел клинической медицины, изучающий терминальные состояния

- Вариант ответа

отделение многопрофильной больницы

- Вариант ответа

практические действия, направленные на восстановление жизнедеятельности

- Вариант ответа

раздел клинической медицины, изучающий термические состояния

10. Реанимация показана:

- Вариант ответа

в каждом случае смерти больного

- Вариант ответа

только при внезапной смерти молодых больных

- Вариант ответа

при внезапно развившихся терминальных состояниях

- Вариант ответа

только при внезапной смерти детей

11. Максимальная продолжительность клинической смерти в обычных условиях составляет:

- Вариант ответа

10-15 мин

- Вариант ответа

5-6 мин

- Вариант ответа

2-3 мин

12. Необходимыми условиями при проведении искусственной вентиляции легких являются:

- Вариант ответа

применение воздуховода

- Вариант ответа

достаточный объем вдуваемого воздуха

- Вариант ответа

валик под лопатками больного

- Вариант ответа

применение воздуховода

13. Первая медицинская помощь при сдавлении конечностей:

- Вариант ответа

новокаиновая блокада, транспортная иммобилизация, введение обезболивающих, сердечных, антигистаминных и противобактериальных средств, прием внутрь соды, ингаляции кислорода

- Вариант ответа

введение промедола, прием внутрь противобактериального средства, тугое бинтование придавленных конечностей от периферии к центру, охлаждение конечности, транспортная иммобилизация

- Вариант ответа

прием внутрь соды и утоление жажды, инъекция атропина, морфия, кофеина и димедрола

- Вариант ответа

иммобилизация транспортными шинами, быстрая эвакуация

личии у них травм позвоночника транспортируются в положении:

- Вариант ответа

на боку на обычных носилках

- Вариант ответа

на животе на обычных носилках

- Вариант ответа

на боку на щите

- Вариант ответа

на спине на щите

14. Шок – это:

- Вариант ответа

острая сердечная недостаточность

- Вариант ответа

острая сердечно-сосудистая недостаточность

- Вариант ответа

острое нарушение периферического кровообращения

- Вариант ответа

острая легочно-сердечная недостаточность

15. При болевом шоке первой развивается:

- Вариант ответа

торпидная фаза шока

- Вариант ответа

эректильная фаза шока

- Вариант ответа

фаза сопротивления

- Вариант ответа

фаза истощения

16. Первое действие при оказании ПМП при синдроме длительного сдавления:

- Вариант ответа

обезболить, наложить жгут

- Вариант ответа

освободить конечность

- Вариант ответа

наложить асептическую повязку

- Вариант ответа

транспортная иммобилизация

17. При сдавливании конечности в течение 4-7 часов возникает:

- Вариант ответа

легкая степень СДР

- Вариант ответа

тяжелая степень СДР

- Вариант ответа

крайне тяжелая степень СДР

- Вариант ответа

средняя степень СДР

18. Какая повязка накладывается при растяжении голеностопных связок:

- Вариант ответа

черепашья
- Вариант ответа
восьмиобразная
- Вариант ответа
уздечка
- Вариант ответа
Колосовидная

19. Какая повязка накладывается при ранениях волосистой части головы:

- Вариант ответа
Восьмиобразная
- Вариант ответа
Т-образная
- Вариант ответа
"Чепец"
- Вариант ответа
Уздечка

20. Какая повязка накладывается при обширных повреждениях груди:

- Вариант ответа
окклюзионная
- Вариант ответа
спиральная
- Вариант ответа
восьмиобразная
- Вариант ответа
Колосовидная

21. При повреждении плеча показана повязка:

- Вариант ответа
колосовидная
- Вариант ответа
восьмиобразная
- Вариант ответа
черепашья
- Вариант ответа
спиральная

22. При обширных отморожениях конечностей используют:

- Вариант ответа
рыцарскую перчатку
- Вариант ответа
варежку
- Вариант ответа

термоизолирующую повязку

- **Вариант ответа**

асептическую

23. Смешанное кровотечение:

- **Вариант ответа**

при одновременном ранении артерий и вен

- **Вариант ответа**

при ранениях вен и капилляров

- **Вариант ответа**

при ранении вен

- **Вариант ответа**

при ранении артерий

24. Первая медицинская помощь при ранении наружной сонной артерии:

- **Вариант ответа**

пальцевое ее прижатие

- **Вариант ответа**

прошивание раны

- **Вариант ответа**

наложение давящей повязки

- **Вариант ответа**

наложение стерильной повязки

25. Первая медицинская помощь при закрытых переломах костей конечностей:

- **Вариант ответа**

наложение транспортной шины

- **Вариант ответа**

транспортировка без транспортной иммобилизации

- **Вариант ответа**

транспортировка пешком

- **Вариант ответа**

транспортировка в «позе лягушки»

26. Мероприятия первой медицинской помощи, проводимые пострадавшему с ожогами:

- **Вариант ответа**

промывание ожоговой поверхности

- **Вариант ответа**

обезболивание

- **Вариант ответа**

инфузионная терапия

- **Вариант ответа**

наложение клеоловой повязки

27. Общие принципы неотложной помощи при отравлениях:

- **Вариант ответа**

вызывание рвоты различными методами, зондовое промывание желудка, стимуляция мочеотделения, удаление неабсорбированных ядов,

- **Вариант ответа**

прекращение дальнейшего поступления яда в организм, применение антидота, восстановление и поддержание нарушенных функций организма, устранение отдельных симптомов интоксикации

- **Вариант ответа**

удаление неабсорбированных токсических веществ, форсированный диурез, гемодиализ, слабительные средства, гемоперфузия, полная санитарная обработка, применение антидота

- **Вариант ответа**

антибиотикотерапия, применение противосудорожных средств

28. Территория, на которой произошел выброс ядовитого вещества в окружающую среду и продолжается его испарение в атмосферу, называется:

- **Вариант ответа**

очагом экологического заражения

- **Вариант ответа**

очагом химического заражения

- **Вариант ответа**

зоной химического заражения

- **Вариант ответа**

зоной экологического заражения

29. Территория, подвергнутая воздействию паров ядовитого вещества, называется:

- **Вариант ответа**

очагом химического заражения

- **Вариант ответа**

зоной экологического заражения

- **Вариант ответа**

зоной химического заражения

- **Вариант ответа**

очагом экологического заражения

30. Промывание желудка при отравлениях кислотами и щелочами производится:

- **Вариант ответа**

после обезболивания рефлекторным методом

- **Вариант ответа**

противопоказано

- Вариант ответа

после обезболивания зондовым методом

- Вариант ответа

после обезболивания физиологическим методом

31. Промывание желудка при отравлениях кислотами и щелочами производится:

- Вариант ответа

нейтрализующими растворами

- Вариант ответа

холодной водой

- Вариант ответа

водой комнатной температуры

- Вариант ответа

теплой водой

32. Наиболее эффективно удаляется яд из желудка:

- Вариант ответа

холодной водой

- Вариант ответа

горячей водой

- Вариант ответа

при промывании рефлекторным методом

- Вариант ответа

при промывании зондовым методом

33. При наличии в атмосфере паров хлора необходимо перемещаться:

- Вариант ответа

в верхние этажи зданий

- Вариант ответа

на улицу

- Вариант ответа

в нижние этажи и подвалы

- Вариант ответа

на крышу

34. При наличии в атмосфере паров хлора дыхательные пути нужно защитить:

- Вариант ответа

ватно-марлевой повязкой, смоченной в растворе питьевой соды

- Вариант ответа

ватно-марлевой повязкой, смоченной в растворе уксусной кислоты

- Вариант ответа

ватно-марлевой повязкой, смоченной кипяченой водой

- Вариант ответа

сухой ватно-марлевой повязкой

35. Пары хлора и аммиака вызывают:

- **Вариант ответа**

возбуждение и эйфорию

- **Вариант ответа**

раздражение верхних дыхательных путей

- **Вариант ответа**

слезотечение

- **Вариант ответа**

Ларингоспазм

36. Способы защиты пищевых продуктов от заражения, загрязнения* при применении оружия массового поражения:

- **Вариант ответа**

герметизация складов и других хранилищ пищевых продуктов, дезинфекция

- **Вариант ответа**

автоклавирование посуды

- **Вариант ответа**

расфасовка пищевых продуктов в герметическую тару, строительство объектов пищевого надзора за городом, герметизация складов и других хранилищ пищевых продуктов

- **Вариант ответа**

ассредоточение пищевых продуктов и строительство объектов пищевого надзора за городом, кипячение

37. Средства обеззараживания воды в очагах массового поражения:

- **Вариант ответа**

гиперхлорирование (с последующим дехлорированием), кипячение, фильтрация, отстаивание, применение перекиси водорода, пергидроля, пантоцида

- **Вариант ответа**

хлорирование, фильтрация, применение перекиси водорода

- **Вариант ответа**

хлорирование, использование пергидроля, пантоцида

- **Вариант ответа**

кипячение, фильтрация, применение перекиси водорода

5. ПОДГОТОВКА К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

При подготовке к зачету по дисциплине «Основы военной подготовки» обучающемуся рекомендуется:

1. повторить пройденный материал и ответить на вопросы, используя конспект и материалы лекций. Если по каким-либо вопросам у студента недостаточно информации в лекционных материалах, то необходимо получить ин-

формацию из раздаточных материалов и/или учебников (литературы), рекомендованных для изучения дисциплины «*Основы военной подготовки*».

Целесообразно также дополнить конспект лекций наиболее существенными и важными тезисами для рассматриваемого вопроса;

2. при изучении основных и дополнительных источников информации в рамках выполнения заданий на *зачете* особое внимание необходимо уделять схемам, рисункам, графикам и другим иллюстрациям, так как подобные графические материалы, как правило, в наглядной форме отражают главное содержание изучаемого вопроса;

3. при изучении основных и дополнительных источников информации в рамках выполнения заданий на *зачете* (в случаях, когда отсутствует иллюстративный материал) особое внимание необходимо обращать на наличие в тексте словосочетаний вида «во-первых», «во-вторых» и т.д., а также дефисов и перечислений (цифровых или буквенных), так как эти признаки, как правило, позволяют структурировать ответ на предложенное задание.

Подобную текстовую структуризацию материала слушатель может трансформировать в рисунки, схемы и т. п. для более краткого, наглядного и удобного восприятия (иллюстрации целесообразно отразить в конспекте лекций – это позволит оперативно и быстро найти, в случае необходимости, соответствующую информацию);

4. следует также обращать внимание при изучении материала для подготовки к *зачету* на словосочетания вида «таким образом», «подводя итог сказанному» и т.п., так как это признаки выражения главных мыслей и выводов по изучаемому вопросу (пункту, разделу). В отдельных случаях выводы по теме (разделу, главе) позволяют полностью построить (восстановить, воссоздать) ответ на поставленный вопрос (задание), так как содержат в себе основные мысли и тезисы для ответа.

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЗАЧЕТУ

1. Каковы виды стрелкового оружия?
2. Какие бывают боеприпасы?
3. Назовите марки ручных гранат.
4. Из чего состоит организационно-штатная структура общевойсковых подразделений?
5. Перечислите Тактико-технические характеристики (ТТХ) основных образцов вооружения и техники ВС РФ.
6. Каковы основные факторы, определяющие характер, организацию и способы ведения современного общевойскового боя?
7. Перечислите основные инженерно-технических мероприятия.
8. Какие существуют защитные сооружения?
9. Какие бывают виды заграждений?
10. На чем основывается полевое водоснабжение?
11. Каковы емкости РДВ?

12. Назовите назначение ТУФ-200.
13. Назовите назначение МТК.
14. Перечислите ТТХ и ТТД вооружения и боевой техники армии США.
15. Перечислите ТТХ и ТТД вооружения и боевой техники армии Германии.
16. Перечислите общие сведения о ядерном оружии.
17. Перечислите общие сведения о химическом оружии
18. Перечислите общие сведения о биологическом оружии
19. Каковы правила поведения и меры профилактики в условиях радиоактивного заражения?
20. Каковы правила поведения и меры профилактики при применении отравляющих веществ?
21. Каковы правила поведения и меры профилактики в условиях применения бактериальных средств?
22. Какие существуют индивидуальные средства РХБ защиты?
23. Каковы мероприятия радиационной, химической и биологической защиты?
24. Каковы тактические свойства местности, их влияние на действия подразделений в боевой обстановке?
25. Опишите назначение, номенклатура и условные знаки топографических карт.
26. Назовите способы ориентирования на местности по карте и без карты.
27. Что такое номенклатура топографических карт?
28. Как задаются координаты объекта?
29. Что такое уточнение координат по "улитке"?
30. Каковы тенденции и особенности развития современных международных отношений?
31. Назовите место и роль России в многополярном мире.
32. Перечислите основные направления социально-экономического развития России.
33. Перечислите основные направления политического развития России.
34. Перечислите основные направления военно-технического развития России.
35. Какие существуют основные положения Военной доктрины РФ?
36. Назовите правовое положение и порядок прохождения военной службы.
37. Что значит нормативно-правовой акт?
38. Чем определяется порядок прохождения военной службы?
39. Назовите основные задачи укрепления безопасности страны.
40. Чем актуальны положения военной доктрины?
41. Перечислите основные тенденции развития военно-политической обстановки.
42. Какие существуют основные требования и категории военной доктрины России?
43. Как взаимосвязаны военная безопасность и жизненно важные интересы?

44. Напишите методологическое значение определения жизненно важных интересов.
45. Перечислите военно-политические основы военной доктрины РФ.
46. Перечислите военно-стратегические основы военной доктрины РФ.
47. Перечислите военно-экономические основы военной доктрины РФ.
48. Перечислите военно-технические основы военной доктрины РФ.
49. Назовите роль и место вооруженных сил в демократическом государстве.
50. Перечислите особенности гражданского контроля за вооруженными силами в демократических государствах.
51. Какие вы знаете особенности дисциплинарной практики?
52. Что такое «статус военнослужащего»?

6. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№ п/п	Наименование	Кол-во экз.
1	Общевойские уставы Вооруженных сил Российской Федерации : курс лекций / составители В. А. Борисов, И. Е. Акулов, В. К. Фоменко. — Томск : Томский политехнический университет, 2019. — 87 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/106173.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
2	Основы огневой подготовки : учебное пособие / А. В. Рыжов, В. М. Коняев, С. В. Пожидаев, Д. В. Горденко. — Москва : Ай Пи Ар Медиа, 2021. — 110 с. — ISBN 978-5-4497-1170-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/109245.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/109245	Эл. ресурс
3	Огневая подготовка : учебное пособие / В. В. Белевцев, Д. В. Горденко, Д. Н. Резеньков, Е. В. Кособлик. — Москва : Ай Пи Ар Медиа, 2021. — 132 с. — ISBN 978-5-4497-1289-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/109244.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/109244	Эл. ресурс
4	Общевойская подготовка. В 2 частях. Ч.1 : учебное пособие / А. Г. Борисов, К. В. Анистратенко, Е. Ю. Лубашев [и др.] ; под редакцией А. Г. Борисова. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2022. — 414 с. — ISBN 978-5-9275-4192-8 (ч.1), 978-5-9275-4191-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/127091.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей. -	Эл. ресурс
5	Общевойская и тактическая подготовка : учебное пособие / С. А. Чеховский, В. Н. Алёшичев, А. С. Евтехов, С. К. Бушанский. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2021. — 280 с. — ISBN 978-5-7433-3472-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/124344.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей. - DOI:	Эл. ресурс

	https://doi.org/10.23682/124344	
6	Баранов, А. Р. Военная топография в служебно-боевой деятельности оперативных подразделений : учебник для курсантов и слушателей военных учебных заведений / А. Р. Баранов, Ю. Г. Маслак, В. И. Ягодинцев. — Москва : Академический проект, 2020. — 159 с. — ISBN 978-5-8291-2944-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/110047.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей.	Эл. ресурс
7	Оказание первой доврачебной помощи в образовательных организациях : учебно-методическое пособие / Ю. В. Азизова, С. К. Касимова, А. В. Трясучев [и др.]. — Астрахань : Астраханский государственный университет, Издательский дом «Астраханский университет», 2020. — 70 с. — ISBN 978-5-9926-1188-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/108843.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
8	Маслова, Л. Ф. Первая помощь пострадавшим : учебное пособие / Л. Ф. Маслова. — Ставрополь : Ставропольский государственный аграрный университет, 2020. — 40 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/121690.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
9	Кутепов, В. А. Тактическая подготовка. Радиационная, химическая и биологическая защита : учебное пособие / В. А. Кутепов, А. Б. Адемченко, С. В. Ковалев. — Омск : Омский государственный технический университет, 2017. — 226 с. — ISBN 978-5-8149-2523-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/78509.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
10	Техническое обеспечение средств радиационной, химической и биологической защиты : учебное пособие / А. В. Шаламов, С. Р. Ахметов, Н. Р. Миннуллин [и др.]. — Казань : Издательство КНИТУ, 2022. — 256 с. — ISBN 978-5-7882-3135-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/129262.html (дата обращения: 25.04.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
11	Боевой устав по подготовке и ведению общевойскового боя. Часть 2. Батальон, рота. — Саратов : Вузовское образование, 2023. — 286 с. — ISBN 978-5-4487-0918-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/127500.html (дата обращения: 23.01.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
12	Боевой устав по подготовке и ведению общевойскового боя. Часть 3. Взвод, отделение, танк. — Саратов : Вузовское образование, 2023. — 224 с. — ISBN 978-5-4487-0917-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/127501.html (дата обращения: 23.01.2023). — Режим доступа: для авторизир. пользователей	Эл. ресурс
13	Баранов, А. Р. Тактико-специальная подготовка войскового разведчика внутренних войск : учебно-практическое пособие / А. Р. Баранов, Ю. Г. Маслак ; под редакцией Ю. Г. Маслак. — Москва : Академический Проект, Трикста, 2015. — 368 с. — ISBN 978-5-8291-1490-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/36874.html (дата обращения: 16.05.2022). — Режим доступа: для авторизир. пользователей	Эл. ресурс