

НЕФТЕЮГАНСКИЙ ИНДУСТРИАЛЬНЫЙ КОЛЛЕДЖ  
(филиал) федерального государственного бюджетного образовательного учреждения  
высшего образования «Югорский государственный университет»

**Методические указания**  
по выполнению практических работ по учебной дисциплине  
**ОП.5 Информационные технологии в**  
**профессиональной деятельности**  
для специальности  
08.02.09 «Монтаж, наладка и эксплуатация электрооборудования  
промышленных и гражданских зданий»

СОГЛАСОВАНО  
Предметной (цикловой)  
комиссией МиЕНД  
Протокол № 1 от 16.09. 2020г.  
Председатель ПЦК  
Шумский Ю.Г.Шумский

УТВЕРЖДЕНО  
заседанием методсовета  
протокол № 1 от 17.09 2020г.  
Председатель методсовета  
Савватеева Н.И.Савватеева

Методические рекомендации по выполнению практических работ по учебной дисциплине Информационные технологии в профессиональной деятельности разработаны на основании рабочей программы учебной дисциплины Информационные технологии в профессиональной деятельности специальности 08.02.09 «Монтаж, наладка и эксплуатация электрооборудования промышленных и гражданских зданий»

Организация-разработчик: Нефтеюганский индустриальный колледж (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Югорский государственный университет»

Разработчик:  
Чупракова И.В. – преподаватель НИК (филиала) ФГБОУ ВО «ЮГУ»

## Содержание

|   |    |
|---|----|
| Пояснительная записка.....  | 4  |
| Перечень практических работ.....  | 6  |
| Практическая работа 1. Моделирование электрических цепей с использованием программы NI Multisim.....                | 7  |
| Практическая работа №2 Практическая работа 2. Построение электрических схем в программе NI Multisim.....            | 14 |
| Практическая работа №3 Изучение виртуальных приборов.....   | 16 |
| Практическая работа № 4. Применение виртуальных приборов для измерения параметров электрических цепей.....          | 21 |
| Практическая работа №5. Применение виртуального осциллографа для изучения переменных сигналов (часть1).....         | 26 |
| Практическая работа №6. Применение виртуального осциллографа для изучения переменных сигналов (часть2).....         | 28 |
| Практическая работа № 7. Моделирование схем.....  | 30 |
| Практическая работа № 8. Моделирование логических схем.....   | 32 |
| Практическая работа № 9. Моделирование схемы электроснабжения квартиры (часть 1).....                               | 35 |
| Практическая работа № 10. Моделирование схемы электроснабжения квартиры (часть2).....                               | 36 |
| Практическая работа №11. Запись математических выражений и вычисление их значений при заданных исходных данных..... | 37 |
| Практическая работа №12. Запись математических выражений и вычисление их значений при заданных исходных данных..... | 42 |
| Практическая работа №13. Работа с комплексными числами в Mathcad (часть 1).....                                     | 45 |
| Практическая работа № 15. Расчет цепей постоянного тока.....  | 48 |
| Практическая работа 16. Сравнение результатов работы в Mathcad с результатами моделирования в Multisim.....         | 50 |
| Практическая работа 17. Расчет цепей переменного тока.....  | 52 |
| Практическая работа18. Сравнение результатов расчетов в Mathcad с результатами моделирования в NI Multisim.....     | 56 |
| Практическая работа 19. Язык программирования C/C++. Идентификаторы.....  | 56 |
| Практическая работа 20. Язык программирования C/C++Операторы. Массивы.....  | 59 |
| Практическая работа 21. Ввод и вывод данных. Первая программ.....   | 61 |
| Практическая работа 22. Условный оператор.....  | 65 |
| Практическая работа 23. Оператор цикла (часть 1).....   | 71 |
| Практическая работа 24. Оператор цикла (часть2).....  | 74 |
| Практическая работа №25. Программирование микроконтроллера на языке C.....  | 78 |
| Практическая работа №26. Программирование микроконтроллеров на языке C.....   | 80 |
| Литература.....   | 84 |

### **Пояснительная записка**

Методические указания по учебной дисциплине «Информационные технологии в профессиональной деятельности» разработаны на основе рабочей программы учебной дисциплины «Информационные технологии в профессиональной деятельности» по специальности 08.02.09 «Монтаж, наладка и эксплуатация электрооборудования промышленных и гражданских зданий».

Учебная дисциплина входит в общепрофессиональный цикл.

Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины

В результате освоения учебной дисциплины обучающийся **должен уметь**:

- пользоваться пакетами специализированных программ для проектирования, расчета и выбора оптимальных параметров систем электроснабжения;
- выполнять расчеты электрических нагрузок;
- выполнять проектную документацию с учетом персонального компьютера.

В результате освоения учебной дисциплины обучающийся **должен знать**:

- пакеты специализированных программ для расчета и проектирования систем электроснабжения;
- о технических решениях по применению микропроцессорной и микроконтроллерной техники в электроэнергетике;
- о программировании микроконтроллеров.

Цель проведения практических работ – уметь:

- Моделировать и строить электрические схемы в программе Multisim;
- Применять виртуальные приборы программы Multisim для измерений в электрической цепи;
- Моделировать и строить логические схемы в программе Multisim;
- Записывать математические выражения и рассчитывать их в программе Mathcad;
- Записывать и рассчитывать комплексные числа в программе Mathcad;
- Рассчитывать электрические цепи в программе Mathcad;
- Сравнить результаты моделирования в программе Multisim и расчетов в программе Mathcad;
- Создавать программы на C/C++;
- Программировать микроконтроллеры.

### **Порядок выполнения работы**

В программе MS Word создать файл: Отчет по выполнению практических работ (ФИО\_группа) в этом отчете записываются все практические работы по мере их выполнения:

- записать название работы, ее цель в «Отчете по выполнению практических работ»;
- выполнить основные задания в соответствии с ходом работы при необходимости скриншоты разместить в «Отчете по выполнению практических работ»;
- выполнить дополнительные задания;
- выводы занести в «Отчет по выполнению практических работ».

### Рекомендации по оформлению практической работы

- При выполнении практических работ в программе Multisim:
  - Проект сохранять в папке «Документы» под именем ФИО\_№пр.работы;
  - Схемы создаются в соответствии с заданием;
  - Проверяются в режиме симулятор;
  - Результаты измерений заносятся в «Отчет по выполнению практических работ».
- При выполнении практических работ в программе Mathcad:
  - Файл задания сохранять в папке «Документы» под именем ФИО\_№пр.работы;
  - Формулы записываются и рассчитываются в соответствии с заданием;
  - Результаты расчетов при необходимости заносятся в «Отчет по выполнению практических работ».
  - При выполнении заданий на сравнение результатов таблицы и скриншоты размещают в «Отчет по выполнению практических работ».
- При выполнении практических работ в C/C++ программы пишутся с сохранением форматирования (отступы, абзацы, специальные символы);
  - Файлы сохраняются папке «Документы» под именем ФИО\_№пр.работы;
  - Скриншоты рабочих программ размещают в Отчет по выполнению практических работ».

Работы проводятся согласно календарно-тематическому планированию, в соответствии с учебной программой.

Пропущенные практические работы выполняются обучающимися самостоятельно и сдаются в отведенные на изучение дисциплины сроки.

При изучении теоретического материала требуется выполнение описанных операций на ПК.

### Критерии оценки работ

**Оценка «отлично»** ставится, если обучающийся выполнил работу в полном объеме с соблюдением необходимой последовательности действий; в «Отчете к практическим работам» правильно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

**Оценка «хорошо»** ставится, если обучающийся выполнил требования к оценке "5", но допущены 2-3 недочета.

**Оценка «удовлетворительно»** ставится, если обучающийся выполнил работу не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки.

**Оценка «неудовлетворительно»** ставится, если обучающийся выполнил работу не полностью или объем выполненной части работы не позволяет сделать правильных выводов.

### Перечень практических работ

| №     | Наименование разделов, тем, занятий   | Кол-во часов |
|-------|---|--------------|
| 1.    | Практическая работа 1. Моделирование электрических цепей с использованием программы NI Multisim               | 2            |
| 2.    | Практическая работа 2. Построение электрических схем в программе NI Multisim.                                 | 2            |
| 3.    | Практическая работа 3. Изучение виртуальных приборов.   | 2            |
| 4.    | Практическая работа 4. Применение виртуальных приборов для измерения параметров электрических цепей.          | 2            |
| 5.    | Практическая работа 5 Применение виртуального осциллографа для изучения переменных сигналов (часть 1).        | 2            |
| 6.    | Практическая работа 6. Применение виртуального осциллографа для изучения переменных сигналов (часть 2).       | 2            |
| 7.    | Практическая работа 7. Моделирование схем.  | 2            |
| 8.    | Практическая работа 8. Моделирование логических схем.   | 2            |
| 9.    | Практическая работа 9. Моделирование схемы электроснабжения квартиры (часть 1)                                | 2            |
| 10.   | Практическая работа 10. Моделирование схемы электроснабжения квартиры (часть 2)                               | 2            |
| 11.   | Практическая работа 11. Запись математических выражений и вычисление их значений при заданных исходных данных | 2            |
| 12.   | Практическая работа 12. Запись математических выражений и вычисление их значений при заданных исходных данных | 2            |
| 13.   | Практическая работа 13. Работа с комплексными числами в Mathcad (часть 1).                                    | 2            |
| 14.   | Практическая работа 14. Работа с комплексными числами в Mathcad (часть 2).                                    | 2            |
| 15.   | Практическая работа 15. Расчет цепей постоянного тока.  | 2            |
| 16.   | Практическая работа 16. Сравнение результатов расчетов в Mathcad с результатами моделирования в NI Multisim.. | 2            |
| 17.   | Практическая работа 17. Расчет цепей переменного тока.  | 2            |
| 18.   | Практическая работа 18. Сравнение результатов расчетов в Mathcad с результатами моделирования в NI Multisim   | 2            |
| 19.   | Практическая работа 19. Язык программирования C/C++. Идентификаторы.  | 2            |
| 20.   | Практическая работа 20. Язык программирования C/C++Операторы. Массивы.  | 2            |
| 21.   | Практическая работа 21. Ввод и вывод данных. Первая программа   | 2            |
| 22.   | Практическая работа 22. Условный оператор.  | 2            |
| 23.   | Практическая работа 23. Оператор цикла (часть 1)  | 2            |
| 24.   | Практическая работа 24. Оператор цикла (часть 2)  | 2            |
| 25.   | Практическая работа 25. Программирование микроконтроллера на языке C.   | 2            |
| 26.   | Практическая работа 26. Программирование микроконтроллера на языке C.   | 2            |
| Итого |   | 52           |

## Практическая работа 1. Моделирование электрических цепей с использованием программы NI Multisim

**ЦЕЛЬ:** Изучить основные принципы измерения информации и кодирование информации.

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

При моделировании электронных схем используются математические модели для воспроизведения поведения реального электронного устройства или схемы. Программное обеспечение для моделирования позволяет моделировать работу схемы и является бесценным инструментом анализа. Благодаря высокоточной способности моделирования многие используют этот тип программного обеспечения в электротехнике, а также программ электронной инженерии

Моделирование поведения схемы перед её фактическим построением может значительно повысить эффективность проектирования, отобразив дефекты конструкции и обеспечив понимание её функционирования. В частности, для интегральных схемоснастка из фотошаблонов является дорогостоящей, макетные платы непрактичны, а зондирование внутренних сигналов чрезвычайно затруднено. Поэтому почти вся конструкция ИС в значительной степени зависит от моделирования. Самый известный симулятор аналоговых схем — SPICE. Самыми известными симуляторы цифровых схем основаны на Verilog и VHDL.

Некоторые электронные симуляторы интегрируют редактор схем, механизм моделирования и экранное отображение формы волны, что позволяет разработчикам быстро модифицировать моделируемую схему и увидеть, как эти изменения влияют на выходные данные. Они также обычно содержат обширные библиотеки моделей и устройств. Эти модели обычно включают в себя специфические для ИС модели транзисторов, общие компоненты, такие как резисторы, конденсаторы, индукторы и трансформаторы, пользовательские модели (такие как управляемые источники тока и напряжения, или модели в Verilog-AMS или VHDL-AMS). Конструкция печатной платы также требует специальных моделей, таких как линии передачи для трассировок и модели IBIS.

В зависимости от способа обработки экспериментальных данных для нахождения результата различают прямые, косвенные, совместные и совокупные измерения.

При *прямом* измерении искомое значение величины находят непосредственно из опытных данных в результате выполнения измерения; например, измерение амперметром тока в ветви цепи.

При *косвенном* измерении искомое значение величины находят на основании известной зависимости между измеряемой величиной и величинами, подвергаемыми прямым измерениям; например, определение сопротивления  $R$  резистора из уравнения  $R = U/I$ , в которое подставляют измеренное значение напряжения  $U$  на зажимах резистора и протекающего через него постоянного тока  $I$ .

Совместные измерения – одновременные измерения нескольких неоднородных величин для нахождения зависимости между ними; например, определение зависимости сопротивления резистора от температуры по формуле  $R_t = R_0(1 + at + bt^2)$  посредством измерения сопротивления резистора  $R_t$  при трех различных температурах  $t$ . Составив систему из трех уравнений, находят параметры  $R_0$ ,  $a$  и  $b$  зависимости сопротивления резистора от температуры.

Совокупные измерения – одновременные измерения нескольких одноименных величин, при которых искомые значения величин находят решением системы уравнений, со-

ставленных из результатов прямых измерений различных сочетаний этих величин; например, определение сопротивлений резисторов, соединенных треугольником, посредством измерения сопротивлений между различными вершинами треугольника. По результатам трех измерений по известным соотношениям определяют сопротивления резисторов треугольника.

Различают также аналоговые и дискретные измерения. При аналоговых измерениях на заданном интервале число измерений электрической величины бесконечно, а при дискретных – число измерений конечно.

В зависимости от способа применения меры известной величины, выделяют при измерениях метод непосредственной оценки и метод сравнения с мерой. При методе непосредственной оценки значение измеряемой величины определяют непосредственно по отсчетному устройству (индикатору) измерительного прибора; например, измерение напряжения с помощью вольтметра. Методы сравнения с мерой – методы, при которых проводится сравнение измеряемой величины и величины, воспроизводимой мерой; например, измерение сопротивления резистора с помощью моста сопротивлений.

Ниже кратко описываются виды и способы измерения электрических величин и параметров компонентов схем электронных устройств с помощью моделей измерительных приборов программной среды интерактивного моделирования и анализа электрических схем NI Multisim (в дальнейшем, для краткости, эту систему будем называть среда MS10). Порядок установки параметров пассивных и активных элементов цепей, измерительных приборов и осциллографа приведен в приложении 2.

#### Измерение тока и напряжения

*Мгновенные* значения напряжения и тока можно измерить с помощью двухканального осциллографа **XSC1**, имитируемого программой MS. Схема подключения осциллографа к цепи рассмотрена в краткой инструкции работы в среде Multisim (см. демонстрационные кадры, запускаемые при щелчке мышью на кнопках **Помощь/Первые шаги в MS10** меню пользовательского интерфейса программного комплекса Labworks).

Измерение *действующих* значений напряжения и тока в ветвях электрической цепи проводится вольтметрами и амперметрами. Амперметр включается *последовательно* с элементами участка цепи, а вольтметр – *параллельно* участку, напряжение на котором необходимо измерить. Модели амперметров и вольтметров среды MS не требуют установки диапазона измерений.

Для установки режима работы и величин внутренних сопротивлений (**Resistance**) амперметров **A1**, **A2** и вольтметров **V1**, **V2** нужно дважды нажать на левую клавишу мыши (в дальнейшем, дважды *щелкнуть мышью*) на изображении соответствующего прибора и в открывшемся диалоговом окне свойств прибора установить в команде **Mode** режим работы (постоянный ток **DC** или переменный **AC**), изменить или оставить установленное по умолчанию внутреннее сопротивление прибора (1 нОм для амперметров и 10 МОм для вольтметров) и нажать на кнопку **OK** (Принять). Внутренние сопротивления 1 нОм для амперметров и 10 МОм для вольтметров, установленные по умолчанию, в большинстве случаев оказывают пренебрежительно малое влияние на работу схем. В библиотеке **Instruments** среды MS имеется мультиметр **XMM1**, используемый для измерения тока, напряжения и сопротивления. В схеме мультиметр, работающий в режиме измерения напряжения, подключается к зажимам резистора **R1** с помощью ключа **S**, управляемого клавишей **S** клавиатуры. В модели мультиметра **XMM1** нужно установить род тока (постоянный "-" или переменный "~"), измеряемую величину по единице измерения: **A** – ток, **V** – напряжение, **Ω** – сопротивление, **dB** – уровень напряжения в децибелах и другие параметры (**SETTINGS**).

В реальных цепях для расширения диапазона измерения тока конкретным амперметром применяют *шунт* (тарированный резистор), включаемый параллельно с амперметром. В этом случае значение измеряемого тока равно показанию амперметра, умноженному на постоянный коэффициент, определяемый по правилу *делителя тока* При измерении боль-



ших переменных токов используют измерительный *трансформатор тока*, первичная обмотка которого включается в ветвь с измеряемым током, а вторичная – замкнута на амперметр. Значение измеряемого тока равно показанию амперметра, умноженную на константу, определяемую коэффициентом трансформации тока измерительного трансформатора.

С целью расширения диапазона измерения напряжения конкретным вольтметром последовательно с его входом включают тарированный резистор. В этом случае значение измеряемого напряжения равно показанию вольтметра, умноженному на коэффициент, определяемый по *правилу делителя напряжения*. При измерении высоких напряжений переменного тока используют измерительный *трансформатор напряжения*, к вторичной обмотке которого подключают вольтметр. Измеряемое напряжение равно показанию вольтметра, умноженному на константу, зависящую от коэффициент трансформации напряжения измерительного трансформатора.

В отдельных случаях при моделировании схем цепей с целью исключения влияния сопротивления амперметра в качестве датчика тока можно использовать идеальный зависимый источник напряжения, управляемый током ИНУТ, а для исключения влияния сопротивления вольтметра при измерении напряжения – идеальный зависимый источник напряжения, управляемый напряжением (ИНУН). Модели зависимых источников энергии размещены в библиотеке **Source** среды MS (примеры применения зависимых источников даны в работах Lr13 и Lr15).

#### Измерение сопротивлений

Для прямого измерения сопротивления резистивного элемента (резистора в том числе) будем использовать мультиметр **XMM2**, в диалоговом окне которого нужно установить режим работы "-" (постоянный ток), измеряемую величину  $\Omega$ , значение тока, например, 10 nA (10 nA) при измерении сопротивлений (**SETTINGS**), и подключить прибор к зажимам отдельного резистора (рис. 1) или параллельно разомкнутому участку резистивной цепи (без источников энергии). При измерении сопротивления между двумя любыми точками схемы цепи, нужно, чтобы хотя бы один из узлов схемы имел соединение с "заземленной" точкой, при этом ветви с идеальными источниками тока должны быть разомкнуты, а идеальные источники напряжения заменены короткозамкнутыми участками (проводниками).

В практике измерения *сопротивлений* резистивных элементов, кроме прямых и сравнительных методов, широко используется так называемый *метод вольтметра-амперметра*, в основу которого положен закон Ома для цепей постоянного тока (рис. 1). Заметим,

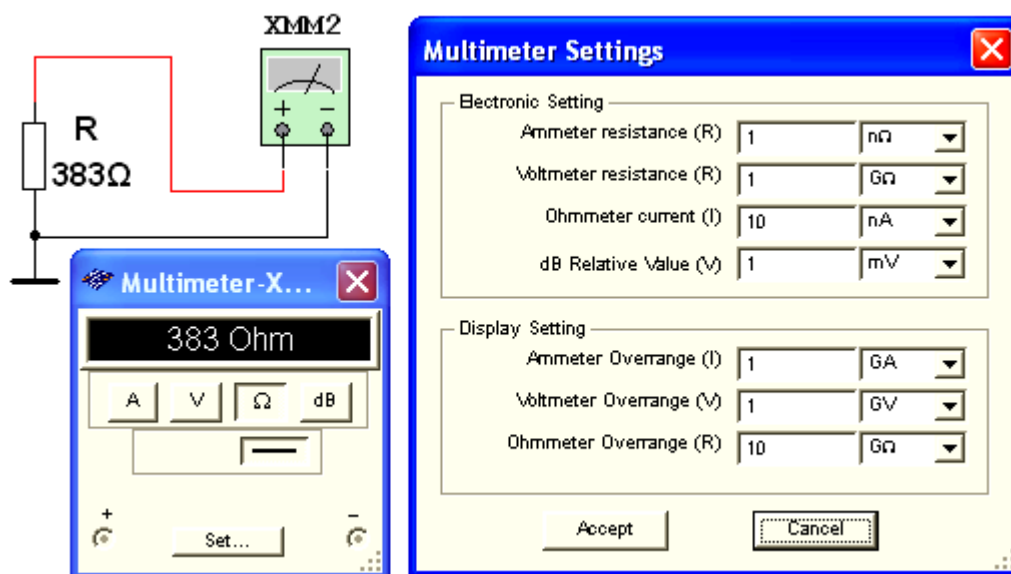


Рисунок 1

что этот метод позволяет получить лишь приближенное значение измеряемого сопротивления  $R \approx U/I$ .

$$R_1 = U/(I - U/R_V),$$

$$R_2 = (U - R_A I)/I,$$

где  $R_V$  и  $R_A$  – внутренние сопротивления вольтметра и амперметра соответственно.

Анализ приведенных выражений позволяет сделать выводы: первой схемой (рис. 1, а) следует пользоваться при измерении сравнительно малых сопротивлений, когда  $R_V \gg R_1$ , а второй схемой – при измерении больших сопротивлений, когда  $R_A \ll R_2$ .

Измерение угла сдвига фаз

Для измерения угла сдвига фаз  $\varphi$  между синусоидальным напряжением и током в реальной цепи используют: измерители разности фаз; так называемый *метод вольтметра-амперметра-ваттметра*, при котором угол  $\varphi$  определяют из уравнения  $\varphi = \arccos(P/UI)$ , где  $P$  – показание ваттметра, а также методы, основанные на измерении временного интервала  $\Delta t$  при помощи электронно-лучевого осциллографа.

Временной интервал

$$\Delta t = \varphi / \omega = \varphi / 2\pi f$$

пропорционален фазовому сдвигу  $\varphi$  между синусоидальным напряжением и током

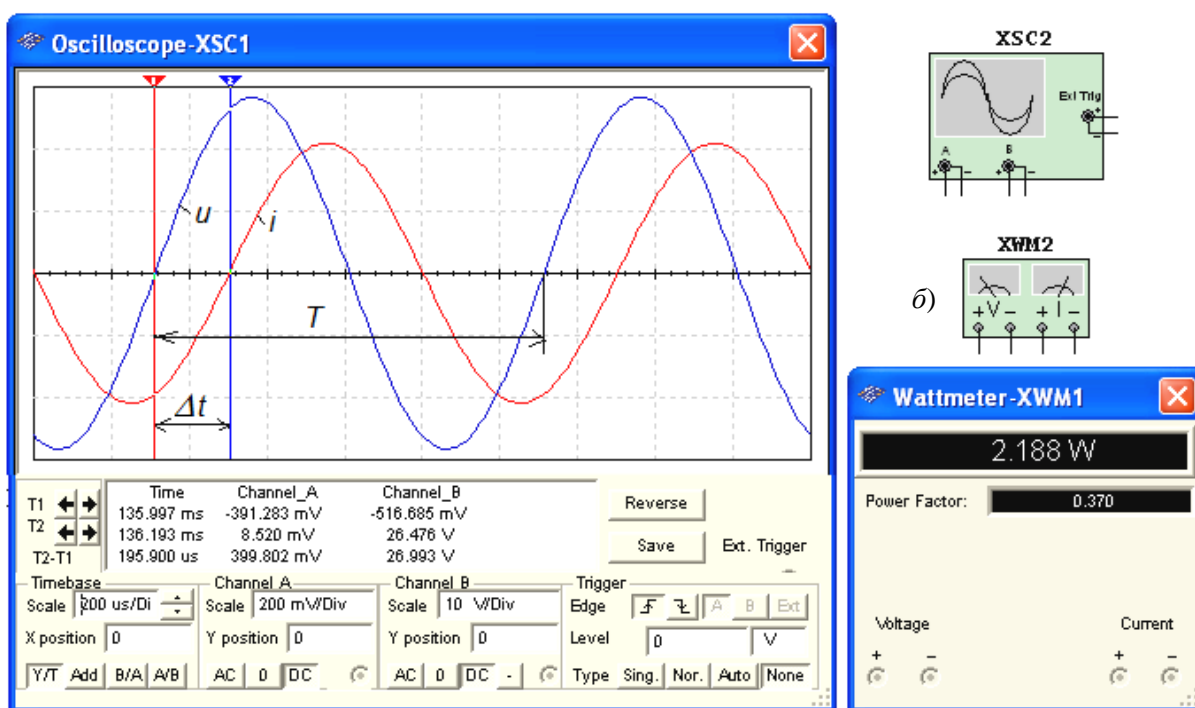


Рисунок 2

в неразветвлённой цепи (рис. 2) и обратно пропорционален угловой частоте  $\omega$  напряжения (тока).

При этом фазовый угол (в градусах) определяют по формуле

$$\varphi = 360^\circ \Delta t / T,$$

где  $T = 1/f$  – период изменения напряжения в секундах (с);  $f$  – частота питающего цепи напряжения в герцах (Гц).

Временной интервал  $\Delta t = T_2 - T_1$  обычно измеряют между нулевыми значениями осциллограмм напряжения и тока с помощью визирных линий (визиров), расположенных слева и справа от экрана осциллографа (рис. 1.3, а). Угол  $\varphi$  берется со знаком "плюс", если ток отстаёт по фазе от напряжения (рис.2), и со знаком "минус", если ток опережает по фазе напряжение.

Установка чувствительности каналов *A* (**Channel A**) и *B* (**Channel B**) и развертки осциллограмм во времени (**Time base**) производится в окне, выводимом ниже поля осциллограмм (рис. 2).

При моделировании схем цепей на рабочем поле программной среды MS10 и их анализе для измерения угла сдвига фаз в цепях переменного тока наряду с осциллографом будем использовать также виртуальный ваттметр **XWM1** (рис.2), размещённый в библиотеке **Instruments**. Ваттметр непосредственно измеряет активную мощность  $P$  цепи (ветви) в ваттах и коэффициент мощности  $\cos \varphi$  (**Power Factor**).

### ХОД РАБОТЫ

**Задание 1.** Изучить краткую инструкцию работы с программной средой NI Multisim (MS), а также мультимедийные кадры с объяснениями, как открыть библиотеку компонентов **Basic**, "перетащить" мышью компоненты на рабочее поле среды MS, соединить их проводниками и установить параметры, изменить цвет проводников; как измерить, с помощью визирных линий осциллографа, временной интервал (сдвиг)  $\Delta t$  между двумя синусоидальными величинами – напряжением и током и рассчитать угол сдвига фаз  $\varphi$  между ними.

**Задание 2.** Открыть библиотеку источников энергии **Source** и "перетащить" на рабочее поле среды MS10 идеальный источник **E1** постоянного напряжения, затем из библиотеки базовых компонентов **Basic** "перетащить" четыре резистора **R1**, ..., **R4**, из библиотеки индикаторов **Indicator** – амперметр **A** и четыре вольтметра **V1**, ..., **V4**, из панели приборов **Instruments** – мультиметр **XMM1**, переключатель **S** из библиотеки **Switch/ SPDT**, управляемый клавишей **S** клавиатуры (английский шрифт).

После двойного щелчка мышью на изображении элемента или прибора в открывающихся диалоговых окнах:

- **здать** ЭДС источника напряжения  $E_1 = N$  (в вольтах), где  $N$  – номер записи фамилии студента в учебном журнале группы;

- **обозначить** (щёлкая мышью на кнопках **Label** и **Value**) резисторы и **установить** значения их сопротивлений:

$$R_1 = N; R_2 = 2N; R_3 = 3N; R_4 = 4N;$$

- **здать** или **оставить** установленный по умолчанию режим **DC** функционирования измерительных приборов и их внутренние сопротивления: 1 нОм для амперметра и 10 Ом для вольтметров;

- **здать** измеряемую величину  $\Omega$  мультиметра **XMM1** и режим его работы (постоянный ток).

**Соединить** параллельно между собой сопротивления  $R_1$  и  $R_3$ ;  $R_2$  и  $R_4$ , и **измерить** с помощью мультиметра **XMM1** сопротивления разветвлений резисторов (рис. 3). Полученные значения сопротивлений **занести** в поля табл. 1 и **сравнить** со значениями, вычисленными по формулам:

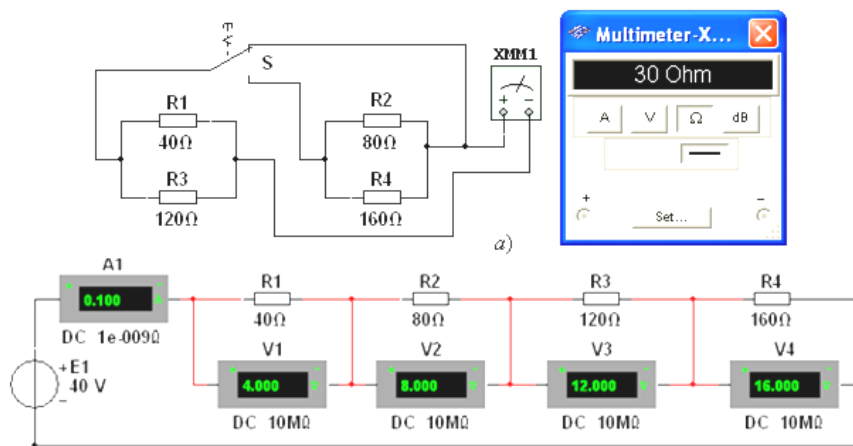



Рисунок 3

$$R_{13} = R_1 R_3 / (R_1 + R_3) \text{ и } R_{24} = R_2 R_4 / (R_2 + R_4).$$

Таблица 1

| Измерено  | $R_{13}$ , Ом | $R_{24}$ , Ом | $U_1$ , мВ | $U_2$ , мВ     | $U_3$ , мВ     | $U_4$ , мВ     |
|-----------|---------------|---------------|------------|----------------|----------------|----------------|
|           |               |               |            | $I_1 = I$ , мА | $I_2 = I$ , мА | $I_3 = I$ , мА |
| Вычислено | $R_{13}$ , Ом | $R_{24}$ , Ом | $R_1$ , Ом | $R_2$ , Ом     | $R_3$ , Ом     | $R_4$ , Ом     |
|           |               |               |            |                |                |                |

**Собрать** схему и, согласно варианту (рис.3), **установить** значения параметров элементов схемы и приборов. **Запустить** программу MS10 (щёлкнуть мышью на кнопке  меню среды MS10) и **занести** показания приборов (значение тока (**A1**) и значения напряжений (**V1**, ..., **V4**) на зажимах резисторов) в табл. 1.

**Рассчитать** сопротивления резисторов и занести их значения в табл. 1.

**Скопировать** и **занести** на страницу электронного отчёта копии рисунков схем (рис.3) (в виде скриншотов после корректировки, например, в редакторе **Paint** или непосредственно после выделения схем и нажатия клавиш **Alt+PrtSc** клавиатуры).

**Задание 3.** Измерить индуктивность катушки и ёмкость конденсатора косвенным методом по результатам прямых измерений напряжения, тока и мощности  $RL$ - или  $RC$ -ветви и косвенного измерения угла сдвига фаз  $\varphi_k$ .

С этой целью **собрать** на рабочем поле программной среды MS1 схему цепи (рис. 4), либо **открыть** файл **1.5.MS10**, и **установить**:

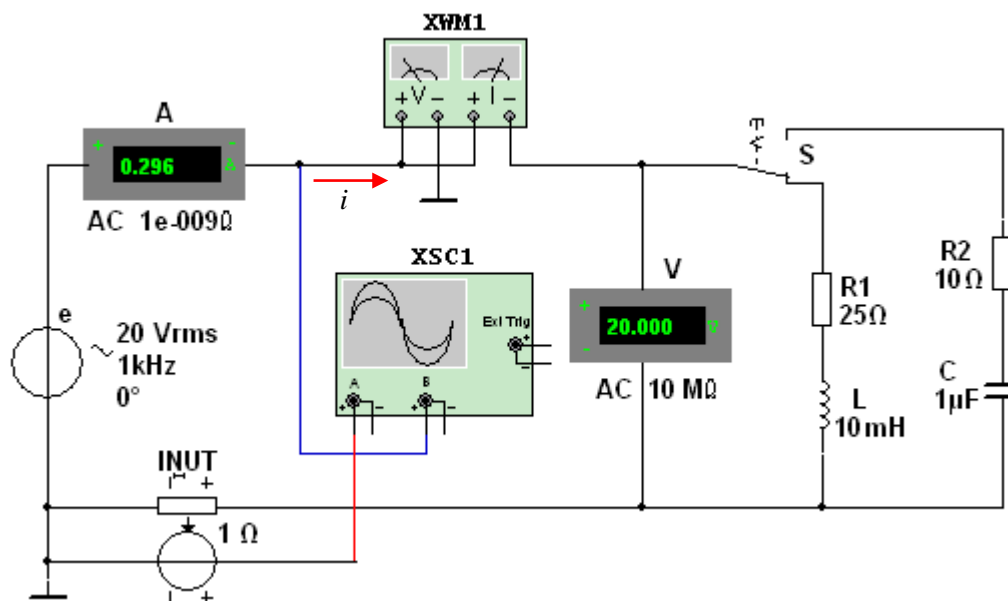


Рисунок 4

– параметры идеального источника синусоидального напряжения  $e = E_m \sin(\omega t + \Psi_u) = \sqrt{2} E \sin(2\pi f t + \Psi_u)$ : действующее значение ЭДС  $E = 5 + N$ , В; частоту  $f = 1$  кГц при измерении индуктивности  $L$  катушки и частоту  $f = 10$  кГц при измерении ёмкости  $C$  конденсатора; начальную фазу напряжения  $\Psi_u = 0$ ;

– режим работы **AC** (переменный ток) амперметра **A** ( $R_A = 1$  нОм) и вольтметра **V** ( $R_V = 10$  МОм);

– значение сопротивления  $R_1 = 25$  Ом резистора **R1** (имитирующего активное сопротивление катушки) и сопротивление  $R_2 = 10$  Ом резистора **R2**;

– значение индуктивности катушки  $L = 5 + \text{int}(N/5)$ , мГн и ёмкости конденсатора  $C = 1 + \text{int}(N/10)$ , мкФ, где  $\text{int}(a/b)$  – целая часть операции  $a/b$ ;

– красный цвет провода, соединённого с каналом **A**, и синий цвет провода, соединённого с каналом **B** двухканального осциллографа **XSC1**;

– **задать** параметры осциллографа **XSC1**. При этом на вход канала **A** подано напряжение с источника **INUT**, пропорциональное входному току  $i$ , а на вход канала **B** подано напряжение  $u$  с зажимов источника напряжения **e**. Цветовая окраска осциллограмм (см. рис. 1.3, *a*) соответствует установленным цветам проводов, соединённых с соответствующими входами каналов прибора **XSC1**.


**Примечание.** В модели осциллографа **XSC1** среды MS0 реализовано "внутреннее соединение" правых входов каналов **A** и **B** (рис.4) с узлом схемы цепи, к которому подключен элемент  $\perp$ , названный "Аналоговая земля". Поэтому не обязательно соединять проводниками правые выводы каналов **A** и **B** осциллографа с "заземлённым" узлом схемы. Элемент  $\perp$  имеет нулевой потенциал. Потенциалы других узлов схемы измеряются относительно заземлённого узла;

– чувствительность 200 мВ/дел (mV/div) канала **A** осциллографа и 5 или 10 В/дел (V/div) канала **B**; длительность развертки (**TIME BASE**) в режиме  $Y/T - 0,2$  мс/дел (2 ms/div). При измерениях указанные цены делений рекомендуется изменять таким образом, чтобы амплитуды напряжений были бы равны не менее 0,5...0,75 высоты экрана осциллографа, а по оси времени укладывалось бы два-три периода колебания напряжений;

– управляющую переключателем клавишу **S** клавиатуры;

– значение коэффициента передачи **INUT**  $k = 1$  Ом;

– управляемый контакт переключателя **S** в нижнее положение, т. е. **подключить**  $R_1L$ -ветвь к источнику **e**.

**Запустить** программу MS (щёлкнуть мышью на кнопке  меню среды MS), **снять** показания приборов и **занести** их в поля табл. 1.2 электронной тетради по работе. Методика определения угла сдвига фаз  $\varphi$  между напряжением и током описана в п. 4 раздела "Теоретические сведения ...".

**Убедиться** (см. рис. 1.3, *a*), что ток  $i_1$  в  $R_1L$ -ветви *отстаёт* по фазе от напряжения  $u$  на угол  $\varphi_1 = \text{arctg}(X_L/R_1) = \text{arccos}(P_1/UI_1)$ , где  $P_1$  – показание ваттметра. **Скопировать** и **занести** на страницу электронной тетради копию рисунка-чертёжа смоделированной схемы рис.3 с показаниями ваттметра и осциллограммами на экране осциллографа (рис. 4).

**Установить** частоту ЭДС  $f = 10$  кГц источника  $e(t)$  и с помощью переключателя **S** подключить  $R_2C$ -ветвь к источнику **e**. Показания приборов **занести** в табл. 2.

**Убедиться** (анализируя расположение осциллограмм на экране осциллографа), что ток  $i_2$  в  $R_2C$ -ветви *опережает* по фазе напряжение  $u$  на угол  $\varphi_2 = \text{arctg}(-X_C/R_2) = -\text{arccos}(P_2/UI_2)$ , где  $P_2$  – показание ваттметра.

**Рассчитать** полное  $Z = U/I$ , активное  $R = Z\cos\varphi$  и реактивное  $X = Z\sin\varphi$  сопротивления  $R_1L$ - и  $R_2C$ -ветви и занести их в поля табл. 1.2.

Так как индуктивное сопротивление катушки  $X_L = \omega L = 2\pi fL$ , Ом, а ёмкостное сопротивление конденсатора  $X_C = 1/\omega C = 1/2\pi fC$ , Ом, то:

– индуктивность катушки, включенной в  $R_1L$ -ветвь:

$$L = X_L/\omega = X_L/2\pi f, \text{ Гн или } L = 10^3 X_L/2\pi f, \text{ мГн};$$

– ёмкость конденсатора, включенного в  $R_2C$ -ветвь:

$$C = 1/(\omega X_C) = 1/(2\pi f \cdot X_C), \text{ Ф или } C = 10^6/(2\pi f \cdot X_C), \text{ мкФ}.$$

| Ветвь  | Установлено |              | Измерено   |             |             |                     | Вычислено   |             |             |              |              |
|--------|-------------|--------------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|--------------|--------------|
|        | $E$ ,<br>В  | $f$ ,<br>кГц | $U$ ,<br>В | $I$ ,<br>мА | $P$ ,<br>Вт | $\varphi$ ,<br>град | $Z$ ,<br>Ом | $R$ ,<br>Ом | $X$ ,<br>Ом | $L$ ,<br>мГн | $C$ ,<br>мкФ |
| $R_1L$ |             | 1            |            |             |             |                     |             |             |             |              | -----        |
| $R_2C$ |             | 10           |            |             |             |                     |             |             |             | -----        |              |

Вычисленные значения индуктивности  $L$  катушки и ёмкости  $C$  конденсатора занести в табл. 2. Сравнить полученные значения  $R$ ,  $L$  и  $C$  с установленными их значениями в схеме цепи.

## Практическая работа №2 Практическая работа 2. Построение электрических схем в программе NI Multisim

**Цель:** изучить технологию построения электрических схем в программе NI Multisim

Мосты постоянного тока широко применяют для измерения электрических и неэлектрических величин. Они строятся на основе метода сравнения с мерой. Задачи, решаемые при построении мостовых измерителей электрических величин, – это уменьшение или полное исключение погрешности от сопротивлений соединительных проводов, обеспечение напряжения, пропорционального приращению преобразуемого сопротивления относительно его начального значения, снижение выходного сопротивления моста и снижение требований к измерительному устройству.

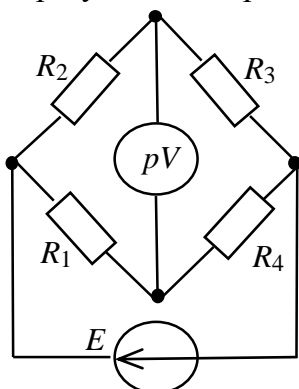


Рисунок 5

Условие равновесия четырехплечего резистивного моста с питанием от источника постоянного напряжения с ЭДС  $E$  (рис 5)

$$R_1R_3 = R_2R_4,$$

откуда величина измеряемого сопротивления, например  $R_2$ ,

$$R_2 = R_1R_3/R_4.$$

Погрешность моста зависит от пределов измерения сопротивлений и обычно указывается в паспорте прибора. В уравновешенном мосте при малом приращении, например

$\Delta R_4$ , напряжение диагонали  $U$  при идеальном источнике напряжения ( $R_{em} = 0$ ) с ЭДС  $E$

$$U \approx E \Delta R_4 R_2 / [(R_1 + R_4)(R_2 + R_3)] \approx E \Delta R_4 / 4R_4,$$

при условии, что  $R_1 = R_2 = R_3 = R_4$ .

Важной характеристикой моста является его чувствительность (в данном случае) по напряжению вблизи его равновесия. Приблизённо абсолютную чувствительность (в мВ/Ом) моста определяют как отношение конечных приращений выходного напряжения и сопротивления, т. е.

$$S_a \approx \Delta U / \Delta R,$$

а относительную чувствительность (в милливольтках) – как отношение приращения выходного напряжения и относительного изменения сопротивления  $\Delta R/R_4$  плеча моста, т. е.

$$S_o = \Delta U / (\Delta R / R_4),$$

где  $\Delta U = U_{n+1} - U_n$  – приращение выходного напряжения в диагонали моста, вызванное изменением сопротивления плеча моста на  $\Delta R = \Delta R_4$ .

**Примечание.** При изменении сопротивлений в двух противоположных плечах моста нужно удвоить величину  $\Delta R$  в формулах абсолютной  $S_a$  и относительной  $S_o$  чувствительностей.

### ХОД РАБОТЫ

**Задание 1.** Собрать на рабочем поле среды MS схему для испытания одинарного резистивного моста (рис. 3), и установить в диалоговых окнах компонентов их параметры или режимы работы:

- резисторов  $R_1 = R_3 = 1 \text{ кОм}$ ;
- потенциометров  $R_2 = R_4 = 1 \text{ кОм}$ ;
- $\Delta R = 0,01N^{1/3} \text{ кОм}$  с округлением третьего знака после запятой – приращение резисторов  $R_2$  и  $R_4$ , где  $N$  – номер записи фамилии студента в учебном журнале группы;
- источника напряжения: ЭДС  $E = 12 \text{ В}$  (источник идеальный,  $R_{em} = 0$ );
- вольтметра **V1**: род тока (**mode**) – постоянный (**DC**), внутреннее сопротивление (**Resistance**)  $R_V = 1 \text{ МОм}$ .

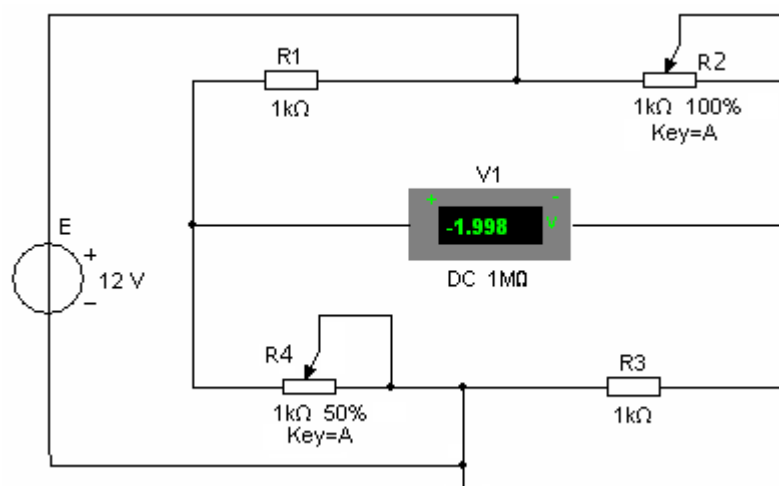


Рисунок 6

Скопировать рис. 6 на страницу отчёта.

**Задание 2.** Снять и построить на одном рисунке характеристики выходного напряжения  $U$  моста от изменения сопротивления одного и двух (противоположных) плеч моста, т. е.  $U(R_4)$  и  $U(R_2, R_4)$ .

С этой целью:

- **рассчитать** и **заполнить** 2-ю и 3-ю колонки табл. 3 и табл. 4 в электронной тетради отчета;
- **запустить** программу (щелкнув мышью в правом верхнем углу меню на цифре 1 кнопки ). **Убедиться**, что мост уравновешен (показание вольтметра **V1** практически равно нулю);
- **остановить** программу (щелкнуть мышью на цифре 0 кнопки ) и **установить** значение сопротивления резистора  $R_4 = R_1 - \Delta R$  ( $n = 1$ );

Таблица 3

| Но-<br>мер<br>опыта | $R_1 =$<br>$R_2 = R_3,$<br>кОм | $R_4 = R_1$<br>$- n\Delta R,$<br>кОм ( $n$<br>$= 0, \dots, 10$ ) | $U,$<br>мВ | $S_{a1},$<br>мВ/Ом | $S_{o1},$<br>мВ |
|---------------------|--------------------------------|--|------------|--------------------|-----------------|
|                     |                                |  |            |                    |                 |

|    |  |  |   |   |   |
|----|--|--|---|---|---|
| 0  |  |  | 0 | – | – |
| 1  |  |  |   |   |   |
| .  |  |  |   |   |   |
| 9  |  |  |   |   |   |
| 10 |  |  |   |   |   |

Таблица 4

| Но-<br>мер<br>опыта | $R_1 = R_3,$<br>кОм | $R_2 = R_4 = R_1 - n\Delta R,$<br>кОм ( $n = 0, \dots, 10$ ) | $U,$<br>мВ | $S_{a2},$<br>мВ/Ом | $S_{o2},$<br>мВ |
|---------------------|---------------------|--|------------|--------------------|-----------------|
| 0                   |                     |  | 0          | –                  | –               |
| 1                   |                     |  |            |                    |                 |
| .                   |                     |  |            |                    |                 |
| 9                   |                     |  |            |                    |                 |
| 10                  |                     |  |            |                    |                 |

– запустить программу, снять показание вольтметра **V1** и занести его в 4-й столбец табл. 4

**Примечание.** Рекомендуется округлять третью или четвертую значащую цифру показания вольтметра;

– повторить опыты при  $n = 2, \dots, 10$  для табл. 2.1, изменяя сопротивление  $R_4$ ;

– повторить опыты при  $n = 1, \dots, 10$  для табл. 2.2, изменяя сопротивления  $R_2$  и  $R_4$ ;

– выбрать масштабы для напряжения  $U$  и сопротивления  $R$  и построить (на одном рисунке) графики  $U(R_4)$  и  $U(R_2, R_4)$ .

**Задание 3.** Рассчитать абсолютную  $S_a$  и относительную  $S_o$  чувствительности мостовой схемы при изменении одного и двух (противоположных) плеч моста, занести их в табл. 3 и в табл. 4 и по данным расчётов построить два графика с двумя характеристиками на каждом:

$S_{a1}(n\Delta R)$  и  $S_{a2}(n \cdot 2\Delta R)$ ;  $S_{o1}(n\Delta R)$  и  $S_{o2}(n \cdot 2\Delta R)$ .

### Практическая работа №3 Изучение виртуальных приборов.

**ЦЕЛЬ:** Изучить виртуальные приборы.

#### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ


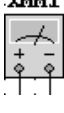

Программа Multisim содержит большое число виртуальных измерительных приборов (инструментов), которые можно использовать с целью проведения измерения или же исследования схемотехнических решений. Виртуальные измерительные приборы по своему действию соответствуют реальным приборам. С их помощью можно не только визуализировать информацию, но и сохранить ее в виде файла данных, который в дальнейшем можно будет использовать для обмена с другими программами, например **LabVIEW**.

Измерительные приборы могут иметь разный внешний вид, в зависимости от того, какую задачу ставит перед собой пользователь и где расположен сам прибор (на панели инструментов или на поле схемы), пример показан в табл. 5.

Представление виртуальных приборов в Multisim



Таблица 5






| Форма представления                      | Описание  | Внешний вид   |
|--|---|---|
| Иконка                                   | Представляет инструмент в панели инструментов Multisim's NI   |  |
| Символ                                   | Представляет инструмент в цепи схемы. Для подсоединения к схеме необходимо использовать внешние выводы инструмента. Для открытия приборной панели необходимо дважды щелкнуть ЛКМ на символе инструмента |  |
| Инструментальная панель (панель прибора) | Открывается двойным щелчком ЛКМ на символе инструмента. Позволяет пользователю взаимодействовать с инструментом – установить параметры измерения<br>Отображает результаты измерения                     |  |





Приборы Multisim позволяют пользователю измерять параметры моделируемой схемы, даже если он не знаком с основами языка моделирования SPICE. Если пользователь изменяет настройки прибора, тут же автоматически изменяются и параметры моделирования.

При проведении моделирования показания приборов постоянно изменяются. В одной и той же схеме может быть несколько экземпляров прибора. Атрибуты настройки прибора и соответствующие этим настройкам параметры моделирования могут быть сохранены в конфигурационном файле. Полученные при моделировании данные при использовании встроенных приборов могут быть обработаны постпроцессором и показаны в окне Grapher View. Внешний вид (размеры) инструментальной панели прибора могут быть изменены в соответствии с требуемым разрешением экрана и способом отображения данных. Данные, полученные в результате анализа, могут быть сохранены в формате файлов TXT, LVM, и TDM.

NI Electronics Workbench Group имеет тесные партнерские связи с представителями ведущих фирм в области измерительной техники, таких Agilent® и Tektronix®, поэтому приборы, размещенные на панели инструментов Multisim, выглядят и работают абсолютно так же, как и реальное физическое приборы этих производителей. Инструменты для анализа напряжения и токов (табл.6)




Таблица 6

| Тип прибора                                     | Функциональные возможности  | Иконка  |
|---|---|---|
| 1   | 2   | 3   |
| Функциональные генераторы (Function generator)  | Генерирование синусоидальных, трапецеидальных и импульсных сигналов. Установка частоты, скважности, амплитуды сигнала |  |
| Мультиметр (Multimeter)                         | Измерение постоянного и переменного тока, напряжения и потерь.  |  |
| 2-канальный осциллограф (Oscilloscope)          | Измерение сигнала в двух каналах. Масштабирование Y и X осей. Смещение по Y оси. Синхронизация                        |  |
| 4-канальный осциллограф (4 channel scilloscope) | Измерение сигнала в четырех каналах. Масштабирование Y и X осей. Смещение по Y оси. Синхронизация                     |  |
| Ваттметр (Wattmeter)                            | Измерение мощности сигнала  |  |

|  |   |   |
|--|---|---|
| Измеритель ВАХ (IV-analysis)                         | Исследуются диоды, биполярные PNP и NPN-транзисторы (BJT). Канальные транзисторы (PMOS), (NMOS) и полевые КМОП структуры (CMOS)                               |  |
| Счетчики (Frequency counter)                         | Измеряются частота, период, фронты импульсов, АЧХ, фазовые сдвиги. Поддерживается частота измерений свыше 10 ГГц, синхронизация, развязка по постоянному току |  |
| Построитель графика Боде (Bode plotter)              | Исследуются частотная характеристика, фазовые сдвиги. Поддерживается частота измерений свыше 10 ГГц   |  |
| Измеритель частотных искажений (Distortion analyzer) | Измеряются интермодуляционные искажения, суммарный коэффициент гармонических искажений (коэффициент гармоник)   |  |



Логические инструменты (табл.7)

Таблица 7

| Тип прибора                            | Функциональные возможности   | Иконка  |
|--|--|---|
| Логический анализатор (Logic analyzer) | Измеряются 16 каналов, история измерений. Поддерживается синхронизация. Внешняя/внутренняя опорная частота   |    |
| Логический конвертер (Logic converter) | Цифровые схемы, построенные по таблицам истинности и логическим выражениям. Таблицы истинности для цифровых схем. Логические выражения для цифровых схем. Реализуются циклы, обновление шага, сброс. HEX, DEC, Boolean, ASCII-коды |   |
| Генератор слов (Word generator)        | Реализуются HEX, DEC, Boolean, ASCII представление данных, синхронизация, временная селекция. Режимы: циклы, обновление шага, сброс  |  |





Приборы радиочастотного диапазона (таб.8)

Таблица 8

| Тип прибора   | Функциональные возможности   | Иконка  |
|---|--|---|
| Анализатор спектра (Spectrum analyzer)                                      | Измеряются спектр, компоненты спектра (мощность, частота), непрерывный и дискретный спектр   |  |
| Прибор для анализа электрических цепей в обобщенном виде (Network analyzer) | Построение по цифровой схеме таблицы истинности или логического выражения. Обратное преобразование таблицы истинности или логического выражения в цифровую схему |  |




Инструменты, моделирующие измерительные приборы фирм-производителей измерительных устройств (таб.9)

Таблица 9

| Тип прибора                                    | Функциональные возможности                                 | Иконка  |
|--|--|---|
| Генератор Agilent (Agilent function generator) | Тип генератора 33120A. Моделирование реального прибора     |  |
| Мультиметр DMM Agilent (Agilent multimeter)    | Тип генератора 34401A. Моделирование реального прибора     |  |
| Осциллограф Agilent (Agilent oscilloscope)     | Тип осциллографа 54622D. Моделирование реального прибора.  |  |
| Осциллограф Tektronix (Tektronix oscilloscope) | Тип осциллографа TDS 2024. Моделирование реального прибора |  |

## Измерительные пробники (таб.10)





Таблица 10

| Тип прибора | Функциональные возможности   | Иконка  |
|-------------|--|---|
| Пробник     | Измерения тока, напряжения и частоты относительно земли            |  |
| Пробник     | Измерения тока, напряжения и частоты относительно другого пробника |  |
| Пробник     | Имитация поведения токовых измерителей (токовых клещей)            |  |

В приложении рассматриваются основы настройки часто используемых виртуальных приборов: генератора слов, логического анализатора, функционального генератора.

Инструменты, базирующиеся на виртуальных приборах NI LabVIEW (таб.11)


Таблица 11

| Имя прибора        | Функциональные возможности                                   | Иконка  |
|--------------------|--|---|
| Микрофон           | Подключение к звуковой плате компьютера. Запись звука        |    |
| Динамик            | Подключение к звуковой плате компьютера                      |    |
| Анализатор сигнала | Анализ сигнала во временной области. Спектр мощности         |    |
| Генератор сигнала  | Гармонический, импульсный, пилообразный, треугольный сигналы |  |

## ХД РАБОТЫ

### Задание 1. Создание проекта и программного файла

При открытии программы Multisim автоматически создается проект схемы под названием Circuit1. Для изменения имени схемы необходимо сохранить его через пункт меню File-Save As, желательно использовать в директории английские буквы. Для создания проекта и программного файла следует выбрать микроконтроллер для программирования. В программе Multisim программируются микроконтроллеры, расположенные в БД в группе MCU, при этом МК выбирается двумя способами:

- через пункт меню Place – Component – MCU – 805x – 8051;
- через панель компонентов:  – 805x – 8051.

Устанавливаем микроконтроллер на рабочей области, появляется всплывающее окно «Мастер по созданию программного файла», которое предлагает выполнить три шага для создания проекта и программного файла.

#### 1. Определение рабочего пространства

В первой строке всплывающего окна указывается путь рабочего пространства для выбранного МК. Используя кнопку «Browse», можно изменить путь рабочего пространства, предложенный программой.

В следующей строке окна предлагается ввести имя рабочего пространства.

#### 2. Создание проекта для микроконтроллера

В этом окне предлагается установить следующие настройки для будущего проекта:

- тип проекта: Standard или Use External Hex File;
- язык программирования: C, Assembly;
- компилятор;
- имя проекта.

### 3. Создание программного файла

В этом окне предлагается создать либо пустой проект, то есть без программного файла, либо добавить исходный программный файл, указав его имя. Работа с Мастером заканчивается нажатием кнопки Finish.

В окне Design Toolbox на закладке Hierarchy возможно просмотреть структуру созданного проекта. Открытие программного файла осуществляется двойным щелчком ЛКМ по его названию в окне Design Toolbox

**Задание 2.** Согласно варианту задания нарисовать схему и указанных элементов (табл. 11). Выполнить соединения элементов (произвольно), ввести позиционные обозначения и нумерацию цепей. При выполнении задания использовать следующие стандарты: ГОСТ 2.702-75 «Правила выполнения электрических схем»; ГОСТ 2.710-81 «Правила выполнения схем». Соединения элементов с МК обозначить зеленым цветом. Также сравнить реальный и виртуальный компоненты, указанные в задании (объяснить, в чем состоит их отличие). Элементы схемы выбрать самостоятельно из базы данных Multisim.

Размещение компонентов производится через пункт меню Place или горячую клавишу Ctrl-W, которые вызывают обращение к проводнику компонентов.

В проводнике компонентов отображается текущая база данных со схемными элементами. В Multisim они организованы в группы (groups) и семейства (families). Также в проводнике компонентов отображается описание компонента (поле Function), модель и печатная плата или производитель.

Для поиска нужного элемента схемы необходимо набрать название компонента в поле Component, и проводник автоматически подберет подходящие элементы. Также требуемый элемент схемы можно найти в соответствующей ему группе (Group). При помощи подменю

«Поиск» (Search) открывается расширенный поиск элементов.

Символ звездочки («\*») в названии компонента заменяет любой набор символов. Например, среди результатов запроса на элемент "74LS\*N" будут микросхемы «74LS01N» и «74LS183N».

При работе с компонентами следует иметь в виду, что любому компоненту соответствует определенная модель в БД, учитывающая различные физические характеристики компонента. Например, операционный усилитель LM358M имеет 5 внешних контактов, но в этой модели БД из них используется только 3, контакты питания не задействованы (неявно заданы). Информация об особенностях используемой модели элемента находится в поле проводника «Производитель/идентификатор» (Model Manuf.\ID), для этого необходимо выделить ЛКМ «Модель» (Model).

Двойной щелчок ЛКМ по компоненту или нажатие кнопки ОК в окне проводника компонентов прикрепит его к курсору. После этого компонент помещается на схему в желаемом месте рабочего пространства при помощи ЛКМ. До установки или после установки элемента в схему его можно повернуть по/против часовой стрелки при помощи горячей клавиши Ctrl-R/Ctrl-R-Shift или выбрать в контекстном меню пункт «90 Clockwise» или «90 Counter CW» соответственно.

Чтобы выбрать компонент на схеме, необходимо щелкнуть по нему ЛК мыши. Для одновременного выбора нескольких компонентов требуется прижать ЛК мыши и перемещать ее, рисуя прямоугольник вокруг нужных компонентов. Выбранные на схеме компоненты выделяются пунктирной линией. Выделение отдельных атрибутов компонента, например, значения или метки, осуществляется одинарным щелчком по соответствующему атрибуту. Клавиша Shift позволяет добавлять или снимать выделение с нескольких компонентов. Выбранные компоненты из БД можно заменить на другие, подобные компоненты с помощью их контекстного меню, пункта Replace Components,

при этом открывается окно проводника компонентов. После замены элемента Multisim восстановит соединения с остальными элементами схемы.

Варианты задания (таб.12)

Таблица 12

| №  | Компоненты схемы   | Элемент для сравнения     |
|----|--|---------------------------|
| 1  | 8-разрядный регистр защелка, биполярный транзистор PNP                 | реле                      |
| 2  | JK-триггер, кнопка   | конденсатор               |
| 3  | EPROM 16Кх8, операционный усилитель                                    | Биполярный транзистор NPN |
| 4  | RAM 2Кх8, конденсатор  | светодиод                 |
| 5  | дешифратор для семисегментного индикатора источник напряжения $V_{cc}$ | катушка индуктивности     |
| 6  | потенциометр, RAM 8Кх8   | диод Шотки                |
| 7  | четыре элемента 2 И-НЕ, светодиод                                      | транзистор биполярный PNP |
| 8  | компаратор, биполярный транзистор NPN                                  | резистор                  |
| 9  | катушка индуктивности, D-триггер                                       | операционный усилитель    |
| 10 | Диодный мост, регистр сдвига   | кварцевый резонатор       |
| 11 | АЦП, пробник   | оптрон                    |
| 12 | 15- сегментный индикатор с общим катодом, предохранитель               | ЦАП                       |

#### Практическая работа № 4. Применение виртуальных приборов для измерения параметров электрических цепей

**ЦЕЛЬ:** Ознакомиться с измерительными приборами, изучить методы и приобрести навыки измерения тока, напряжения, мощности, угла сдвига фаз между синусоидальным напряжением и током, а также сопротивлений резисторов, индуктивностей индуктивных катушек и ёмкостей конденсаторов.

#### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

В зависимости от способа обработки экспериментальных данных для нахождения результата различают прямые, косвенные, совместные и совокупные измерения.

При *прямом* измерении искомое значение величины находят непосредственно из опытных данных в результате выполнения измерения; например, измерение амперметром тока в ветви цепи.

При *косвенном* измерении искомое значение величины находят на основании известной зависимости между измеряемой величиной и величинами, подвергаемыми прямым измерениям; например, определение сопротивления  $R$  резистора из уравнения  $R = U/I$ , в которое подставляют измеренное значение напряжения  $U$  на зажимах резистора и протекающего через него постоянного тока  $I$ .

*Совместные* измерения – одновременные измерения нескольких неоднородных величин для нахождения зависимости между ними; например, определение зависимости сопротивления резистора от температуры по формуле  $R_t = R_0(1 + at + bt^2)$  посредством измерения сопротивления резистора  $R_t$  при трех различных температурах  $t$ . Составив систему из трех уравнений, находят параметры  $R_0$ ,  $a$  и  $b$  зависимости сопротивления резистора от температуры.

*Совокупные* измерения – одновременные измерения нескольких одноименных величин, при которых искомые значения величин находят решением системы уравнений, со-

ставленных из результатов прямых измерений различных сочетаний этих величин; например, определение сопротивлений резисторов, соединенных треугольником, посредством измерения сопротивлений между различными вершинами треугольника. По результатам трех измерений по известным соотношениям определяют сопротивления резисторов треугольника.

Различают также аналоговые и дискретные измерения. При *аналоговых* измерениях на заданном интервале число измерений электрической величины бесконечно, а при *дискретных* – число измерений конечно.

В зависимости от способа применения меры известной величины, выделяют при измерениях метод непосредственной оценки и метод сравнения с мерой. При *методе непосредственной оценки* значение измеряемой величины определяют непосредственно по отсчётному устройству (индикатору) измерительного прибора; например, измерение напряжения с помощью вольтметра. *Методы сравнения с мерой* – методы, при которых проводится сравнение измеряемой величины и величины, воспроизводимой мерой; например, измерение сопротивления резистора с помощью моста сопротивлений.

Ниже кратко описываются виды и способы измерения электрических величин и параметров компонентов схем электронных устройств с помощью моделей измерительных приборов программной среды интерактивного моделирования и анализа электрических схем NI Multisim 10 (в дальнейшем, для краткости, эту систему будем называть среда MS10). Порядок установки параметров пассивных и активных элементов цепей, измерительных приборов и осциллографа приведен в приложении 2.

### Измерение тока и напряжения

*Мгновенные* значения напряжения и тока можно измерить с помощью двухканального осциллографа XSC1, имитируемого программой MS10. Схема подключения осциллографа к цепи рассмотрена в краткой инструкции работы в среде Multisim (см. демонстрационные кадры, запускаемые при щелчке мышью на кнопках **Помощь/Первые шаги в MS10** меню пользовательского интерфейса программного комплекса Labworks).

Измерение *действующих* значений напряжения и тока в ветвях электрической цепи проводится вольтметрами и амперметрами. Амперметр включается *последовательно* с элементами участка цепи, а вольтметр – *параллельно* участку (рис. 1.1, *a* и *б*), напряжение на котором необходимо измерить. Модели амперметров и вольтметров среды MS10 не требуют установки диапазона измерений.

Для установки режима работы и величин внутренних сопротивлений (**Resistance**) амперметров **A1**, **A2** и вольтметров **V1**, **V2** нужно дважды нажать на левую клавишу мыши (в дальнейшем, дважды *щелкнуть мышью*) на изображении соответствующего прибора и в открывшемся диалоговом окне свойств прибора установить в команде **Mode** режим работы (постоянный ток **DC** или переменный **AC**), изменить или оставить установленное по умолчанию внутреннее сопротивление прибора (1 нОм для амперметров и 10 МОм для вольтметров) и нажать на кнопку **OK** (Принять). Внутренние сопротивления 1 нОм для амперметров и 10 МОм для вольтметров, установленные по умолчанию, в большинстве случаев оказывают пренебрежительно малое влияние на работу схем.

В библиотеке **Instruments** среды MS10 имеется мультиметр **XMM1** (рис. 1.1, *a*), используемый для измерения тока, напряжения и сопротивления. В схеме (рис. 1.1, *a*) мультиметр, работающий в режиме измерения напряжения, подключается к зажимам резистора **R1** с помощью ключа **S**, управляемого клавишей **S** клавиатуры. В модели мультиметра **XMM1** нужно установить род тока (постоянный "-" или переменный "~"), измеряемую величину по единице измерения: **A** – ток, **V** – напряжение, **Ω** – сопротивление, **dB** – уровень напряжения в децибелах и другие параметры (**SETTINGS**) (рис. 7).

В реальных цепях для расширения диапазона измерения тока конкретным амперметром применяют *шунт* (тарированный резистор), включаемый параллельно с амперметром. В этом случае значение измеряемого тока равно показанию амперметра, умноженному на постоянный коэффициент, определяемый по правилу *делителя тока*. При измерении больших

переменных токов используют измерительный *трансформатор тока*, первичная обмотка которого включается в ветвь с измеряемым током, а вторичная – замкнута на амперметр. Значение измеряемого тока равно показанию амперметра, умноженную на константу, определяемую коэффициентом трансформации тока измерительного трансформатора.

С целью расширения диапазона измерения напряжения конкретным вольтметром последовательно с его входом включают тарированный резистор. В этом случае значение измеряемого напряжения равно показанию вольтметра, умноженному на коэффициент, определяемый по *правилу делителя напряжения*. При измерении высокого напряжения переменного тока используют измерительный *т*, которого подключают вольтметр. Измеренное значение напряжения умноженному на константу, зависящую от коэффициента трансформации измерительного трансформатора.

**Примечание.** В отдельных случаях для уменьшения влияния сопротивления амперметра на измеряемый ток используют дополнительный источник напряжения, управляемый напряжением, размещены в библиотеке **Source**

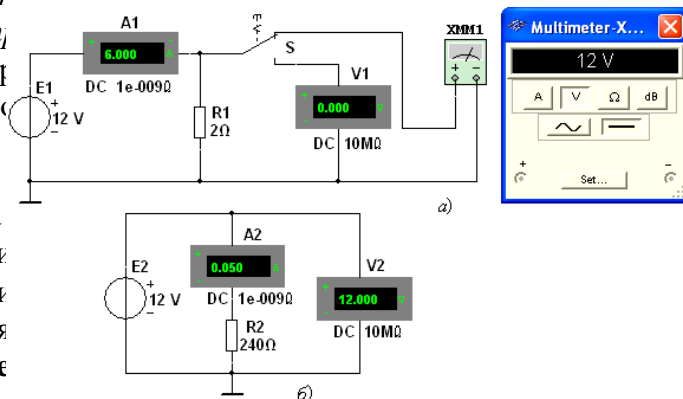


Рисунок 7

### Измерение сопротивлений

Для прямого измерения сопротивления резистивного элемента (резистора в том числе) будем использовать мультиметр **XMM2**, в диалоговом окне которого нужно установить режим работы "-" (постоянный ток), измеряемую величину  $\Omega$ , значение тока, например, 10 nA (10 nA) при измерении сопротивлений (**SETTINGS**), и подключить прибор к зажимам отдельного резистора (рис. 8) или параллельно разомкнутому участку резистивной цепи (без источников энергии). При измерении сопротивления между двумя любыми точками схемы цепи, нужно, чтобы хотя бы один из узлов схемы имел соединение с "заземленной" точкой, при этом ветви с идеальными источниками тока должны быть разомкнуты, а идеальные источники напряжения заменены короткозамкнутыми участками (проводниками).

В практике измерения *сопротивлений* резистивных элементов, кроме прямых и сравнительных методов, широко используется так называемый *метод вольтметра-амперметра*, в основу которого положен закон Ома для цепей постоянного тока. Заметим, что этот метод

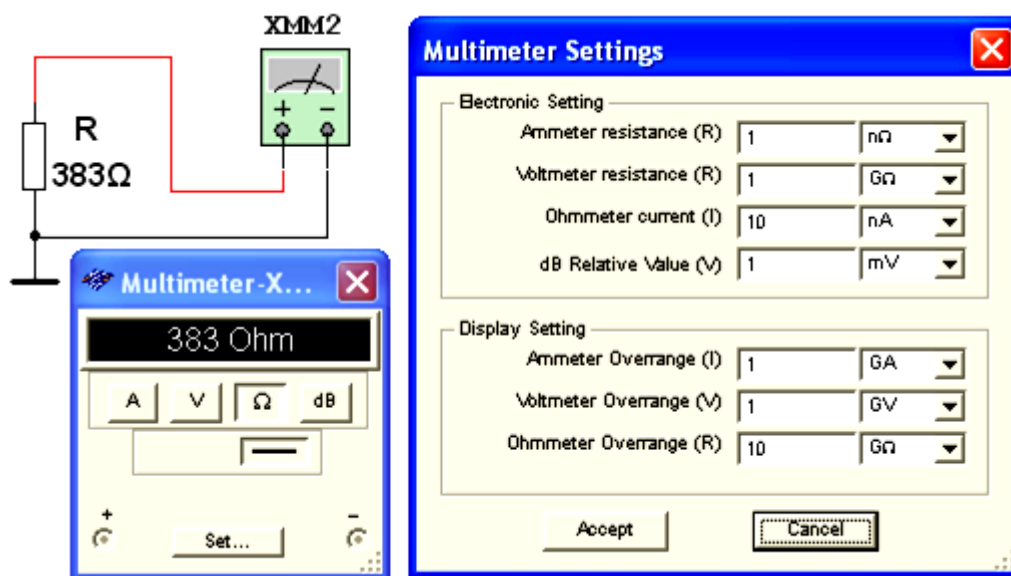


Рисунок 8

позволяет получить лишь приближенное значение измеряемого сопротивления  $R \approx U/I$ .  $R_1 = U/(I - U/R_V)$ ,

$$R_2 = (U - R_A I)/I,$$

где  $R_V$  и  $R_A$  – внутренние сопротивления вольтметра и амперметра соответственно.

Анализ приведенных выражений позволяет сделать выводы: первой схемой (рис. 1.1, а) следует пользоваться при измерении сравнительно малых сопротивлений, когда  $R_V \gg R_1$ , а второй схемой (рис. 1.1, б) – при измерении больших сопротивлений, когда  $R_A \ll R_2$ .

### Измерение угла сдвига фаз

Для измерения угла сдвига фаз  $\varphi$  между синусоидальным напряжением и током в реальной цепи используют: измерители разности фаз; так называемый *метод вольтметра-амперметра-ваттметра*, при котором угол  $\varphi$  определяют из уравнения  $\varphi = \arccos(P/UI)$ , где  $P$  – показание ваттметра, а также методы, основанные на измерении временного интервала  $\Delta t$  при помощи электронно-лучевого осциллографа.

Временной интервал

$$\Delta t = \varphi / \omega = \varphi / 2\pi f$$

пропорционален фазовому сдвигу  $\varphi$  между синусоидальным напряжением и током в не-

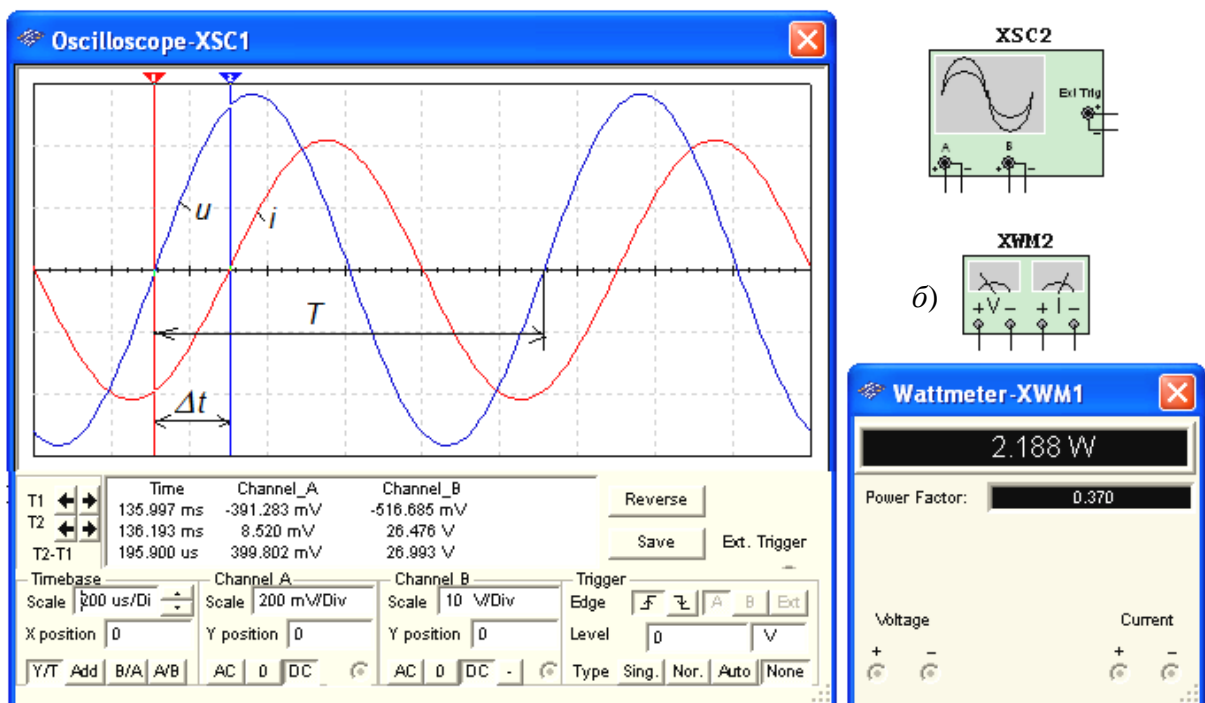


Рисунок 9

разветвлённой цепи (рис. 9) и обратно пропорционален угловой частоте  $\omega$  напряжения (тока).

При этом фазовый угол (в градусах) определяют по формуле

$$\varphi = 360^\circ \Delta t / T,$$

где  $T = 1/f$  – период изменения напряжения в секундах (с);  $f$  – частота питающего цепь напряжения в герцах (Гц).

Временной интервал  $\Delta t = T_2 - T_1$  обычно измеряют между нулевыми значениями осциллограмм напряжения и тока с помощью визирных линий (визиров), расположенных слева и справа от экрана осциллографа (рис. 9). Угол  $\varphi$  берется со знаком "плюс", если ток отстаёт по фазе от напряжения (см. рис. 9), и со знаком "минус", если ток опережает по фазе напряжение.



Установка чувствительности каналов *A* (**Channel A**) и *B* (**Channel B**) и развертки осциллограмм во времени (**Time base**) производится в окне, выводимом ниже поля осциллограмм (см. рис. 9).

При моделировании схем цепей на рабочем поле программной среды MS10 и их анализе для измерения угла сдвига фаз в цепях переменного тока наряду с осциллографом будем использовать также виртуальный ваттметр **XWM1** (рис. 9), размещённый в библиотеке **Instruments**. Ваттметр непосредственно измеряет активную мощность  $P$  цепи (ветви) в ваттах и коэффициент мощности  $\cos \varphi$  (**Power Factor**).

### ХОД РАБОТЫ

**Задание 1.** Изучить краткую инструкцию работы с программной средой NI Multisim (MS), открыть библиотеку компонентов **Basic**, "перетащить" мышью компоненты на рабочее поле среды MS10, соединить их проводниками и установить параметры, изменить цвет проводников; как измерить, с помощью визирных линий осциллографа, временной интервал (сдвиг)  $\Delta t$  между двумя синусоидальными величинами – напряжением и током и рассчитать угол сдвига фаз  $\varphi$  между ними.

**Задание 2.** Открыть библиотеку источников энергии **Source** и "перетащить" на рабочее поле среды MS10 идеальный источник **E1** постоянного напряжения, затем из библиотеки базовых компонентов **Basic** "перетащить" четыре резистора **R1**, ..., **R4**, из библиотеки индикаторов **Indicator** – амперметр **A** и четыре вольтметра **V1**, ..., **V4**, из панели приборов **Instruments** – мультиметр **XMM1**, переключатель **S** из библиотеки **Switch/ SPDT**, управляемый клавишей **S** клавиатуры (английский шрифт).

После двойного щелчка мышью на изображении элемента или прибора в открывающихся диалоговых окнах:

- **здать** ЭДС источника напряжения  $E_1 = N$  (в вольтах), где  $N$  – номер записи фамилии студента в учебном журнале группы;

- **обозначить** (щёлка мышью на кнопках **Label** и **Value**) резисторы и **установить** значения их сопротивлений:

$$R_1 = N; R_2 = 2N; R_3 = 3N; R_4 = 4N;$$

- **здать** или **оставить** установленный по умолчанию режим **DC** функционирования измерительных приборов и их внутренние сопротивления: 1 нОм для амперметра и 10 МОм для вольтметров;

- **здать** измеряемую величину  $\Omega$  мультиметра **XMM1** и режим его работы (постоянный ток).

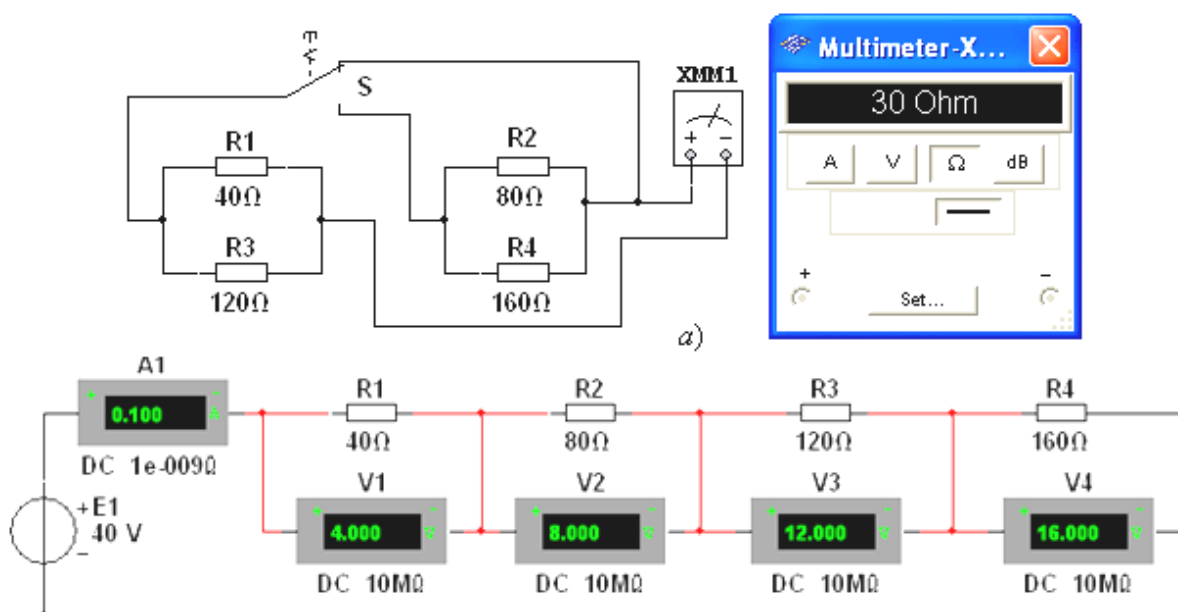



Рисунок 10

**Соединить** параллельно между собой сопротивления  $R_1$  и  $R_3$ ;  $R_2$  и  $R_4$ , и **измерить** с помощью мультиметра **ХММ1** сопротивления разветвлений резисторов (см. рис. 10). Полученные значения сопротивлений **занести** в поля табл. 13 и **сравнить** со значениями, вычисленными по формулам:

$$R_{13} = R_1 R_3 / (R_1 + R_3) \text{ и } R_{24} = R_2 R_4 / (R_2 + R_4).$$

Таблица 13

|           |               |               |                |                |                |                |
|-----------|---------------|---------------|----------------|----------------|----------------|----------------|
| Измерено  | $R_{13}$ , Ом | $R_{24}$ , Ом | $U_1$ , мВ     | $U_2$ , мВ     | $U_3$ , мВ     | $U_4$ , мВ     |
|           |               |               | $I_1 = I$ , мА | $I_2 = I$ , мА | $I_3 = I$ , мА | $I_4 = I$ , мА |
|           |               |               |                |                |                |                |
| Вычислено | $R_{13}$ , Ом | $R_{24}$ , Ом | $R_1$ , Ом     | $R_2$ , Ом     | $R_3$ , Ом     | $R_4$ , Ом     |
|           |               |               |                |                |                |                |

**Собрать** схему и, согласно варианту (см. рис. 10), **установить** значения параметров элементов схемы и приборов. **Запустить** программу MS (щёлкнуть мышью на кнопке  меню среды MS10) и **занести** показания приборов (значение тока (**A1**) и значения напряжений (**V1**, ..., **V4**) на зажимах резисторов..

**Рассчитать** сопротивления резисторов и занести их значения в табл. 12

**Скопировать** и **занести** на страницу электронного отчёта копии рисунков схем (см. рис. 10) (в виде скриншотов после корректировки, например, в редакторе **Paint** или непосредственно после выделения схем и нажатия клавиш **Alt+PrtSc** клавиатуры).

### Практическая работа №5. Применение виртуального осциллографа для изучения переменных сигналов (часть1)

**Цель работы:** научиться применять виртуальный осциллограф в электрической цепи для изучения переменных сигналов.

#### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

В Multisim есть несколько модификаций осциллографов, которыми можно управлять как настоящими. Они позволяют устанавливать параметры временной развертки и напряжения, выбирать тип и уровень запуска измерений. Данные осциллографов можно посмотреть после эмуляции с помощью самописца (Grapher) из меню Вид\Плоттер (View/Grapher).

В Multisim есть следующие осциллографы:

- 2-х канальный
- 4-х канальный
- осциллограф смешанных сигналов Agilent 54622D
- 4-х канальный цифровой осциллограф с записью Tektronix TDS 2024

Осциллограф Multisim имеет три режима синхронизации,

#### **Нормальный, Однократный и Автоматический:**

**Нормальный.** Этот сигнал создается напряжением указанного источника (Канал А, Канал В или Внешним источником) и пересекает уровень запуска. После этого луч перемещается вправо и рисует кривую. Когда он достигает правой части экрана, то возвращается в левую часть и ожидает следующего сигнала синхронизации. В этом режиме осциллограмма на экране неподвижна;

**Однократный.** Данный режим работает аналогично режиму **Норм.**, за исключением того, что на экране формируется только одна кривая. При активации оп-

ции **Однократн.** осциллограф переходит в режим ожидания сигнала синхронизации. После получения сигнала синхронизации рисуется кривая, а потом луч отключается.

Другие кривые не создаются до тех пор, пока пользователь повторно не активирует режим ожидания запуска, а система не получит сигнал запуска. Обычно этот режим используется при работе с одиночными осциллограммами (например с кривыми импульсов, которые не повторяются).

**Автоматический.** В этом режиме сигнал синхронизации создается автоматически, а не путем сравнения заданного значения и уровня триггера. Во многих осциллографах для создания сигнала синхронизации используется напряжение частотой 60 Гц от источника питания. Момент запуска не синхронизируется с измеряемым сигналом, в результате осциллограмма будет перемещаться по экрану. Получаемая в этом режиме осциллограмма никак не связана с моментом запуска. Это значит, что начальная точка кривой является произвольной и постоянно изменяется.

Автоматический режим синхронизации обычно используется при измерении напряжения постоянного тока, поскольку такой сигнал не изменяется и, следовательно, не пересекает уровня запуска синхронизации.

## ХОД РАБОТЫ

Собрать усилитель на [биполярном транзисторе](#) по схеме с общим эмиттером. В качестве источника входного сигнала используем функциональный генератор (function generator). В настройках ФГ выберем синусоидальный сигнал амплитудой 0,1 В, частотой 18,2 кГц.

С помощью осциллографа (oscilloscope) снимем осциллограммы входного и выходного сигналов, для этого нам понадобится задействовать оба канала.

Чтобы проверить правильность показаний осциллографа поставим на вход и на выход по мультиметру, переключив их предварительно в режим вольтметра.

Запускаем схему и открываем двойным кликом каждый прибор рис.11.

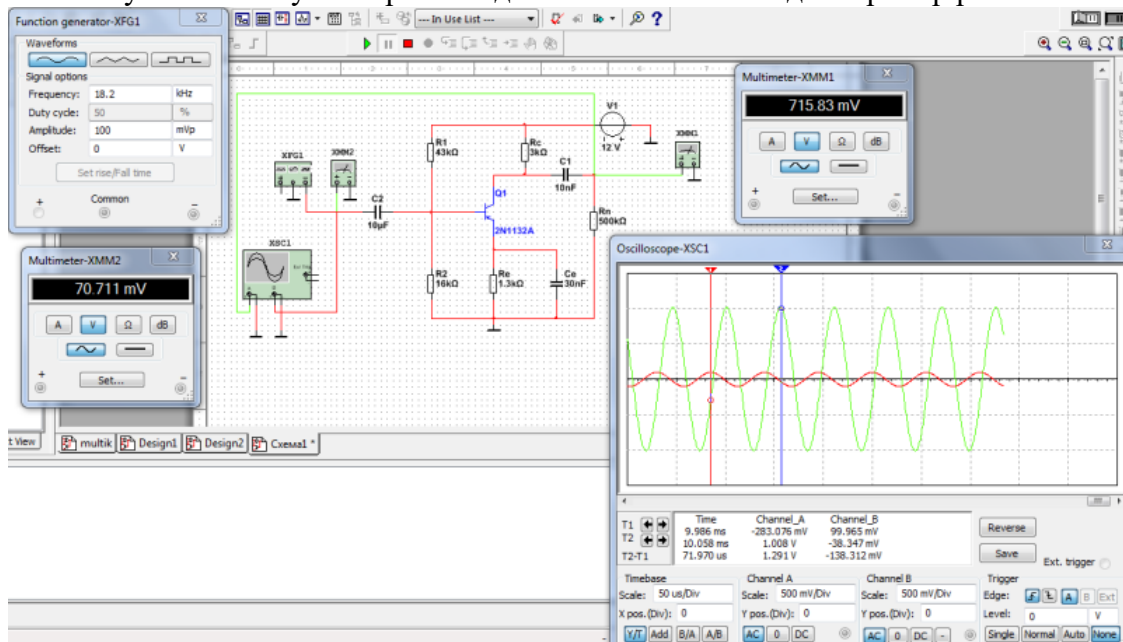


Рисунок 11

Показания вольтметров должны совпадать с показаниями осциллографа, если знать что вольтметр показывает действующее значение напряжения, для получения которого необходимо разделить амплитудное значение на корень из двух.

## Практическая работа №6. Применение виртуального осциллографа для изучения переменных сигналов (часть2)

**Цель работы:** научиться применять виртуальный осциллограф в электрической цепи для изучения переменных сигналов.

**Задание 1.** Измерить индуктивность катушки и ёмкость конденсатора косвенным методом по результатам прямых измерений напряжения, тока и мощности  $RL$ - или  $RC$ -ветви и косвенного измерения угла сдвига фаз  $\varphi_k$ .

С этой целью **собрать** на рабочем поле программной среды MS схему цепи (см. рис. 12)

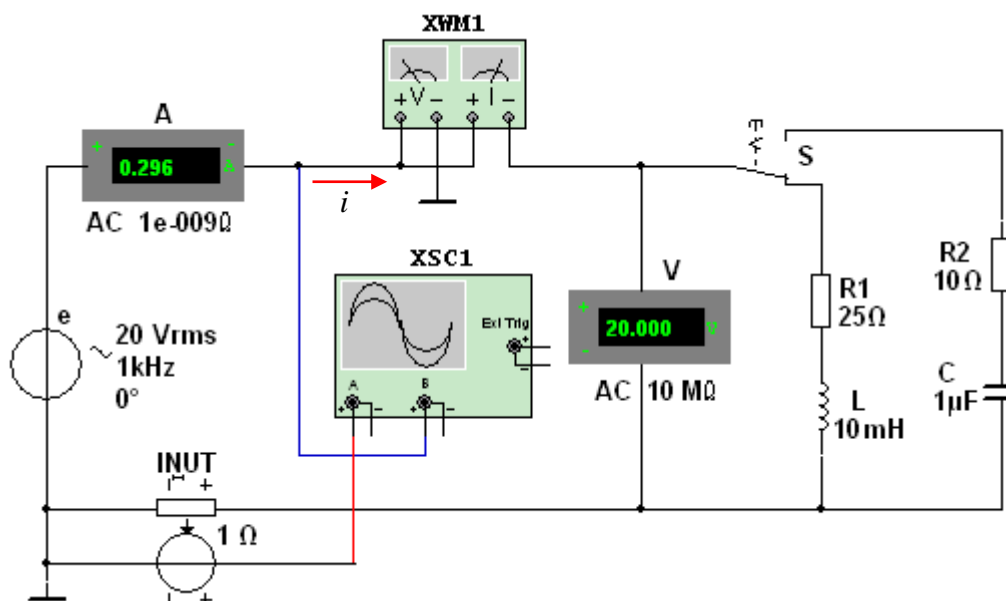


Рисунок 12

– параметры идеального источника синусоидального напряжения  $e = E_m \sin(\omega t + \Psi_u) = \sqrt{2}E \sin(2\pi ft + \Psi_u)$ : действующее значение ЭДС  $E = 5 + N$ , В; частоту  $f = 1$  кГц при измерении индуктивности  $L$  катушки и частоту  $f = 10$  кГц при измерении ёмкости  $C$  конденсатора; начальную фазу напряжения  $\Psi_u = 0$ ;

– режим работы AC (переменный ток) амперметра **A** ( $R_A = 1$  нОм) и вольтметра **V** ( $R_V = 10$  МОм);

– значение сопротивления  $R_1 = 25$  Ом резистора **R1** (имитирующего активное сопротивление катушки) и сопротивление  $R_2 = 10$  Ом резистора **R2**;

– значение индуктивности катушки  $L = 5 + \text{int}(N/5)$ , мГн и ёмкости конденсатора  $C = 1 + \text{int}(N/10)$ , мкФ, где  $\text{int}(a/b)$  – целая часть операции  $a/b$ ;

– красный цвет провода, соединённого с каналом **A**, и синий цвет провода, соединённого с каналом **B** двухканального осциллографа **XSC1**;

– **задать** параметры осциллографа **XSC1**. При этом на вход канала **A** подано напряжение с источника **INUT**, пропорциональное входному току  $i$ , а на вход канала **B** подано напряжение  $u$  с зажимов источника напряжения **e**. Цветовая окраска осциллограмм (см. рис. 1.3, а) соответствует установленным цветам проводов, соединённых с соответствующими входами каналов прибора **XSC1**.

**Примечание.** В модели осциллографа **XSC1** среды MS реализовано "внутреннее соединение" правых входов каналов **A** и **B** с узлом схемы цепи, к которому подключен элемент  $\perp$ , названный "Аналоговая земля". Поэтому не обязательно соединять проводниками


правые выводы каналов **A** и **B** осциллографа с "заземлённым" узлом схемы. Элемент  $\perp$  имеет нулевой потенциал. Потенциалы других узлов схемы измеряются относительно заземлённого узла;

– чувствительность 200 мВ/дел (mV/div) канала **A** осциллографа и 5 или 10 В/дел (V/div) канала **B**; длительность развертки (**TIME BASE**) в режиме  $Y/T$  – 0,2 мс/дел (2 ms/div). При измерениях указанные цены делений рекомендуется изменять таким образом, чтобы амплитуды напряжений были бы равны не менее 0,5...0,75 высоты экрана осциллографа, а по оси времени укладывалось бы два-три периода колебания напряжений;

– управляющую переключателем клавишу **S** клавиатуры;

– значение коэффициента передачи **INUT**  $k = 1$  Ом;

– управляемый контакт переключателя **S** в нижнее положение, т. е. **подключить**  $R_1L$ -ветвь к источнику **e**.

**Запустить** программу MS10 (щёлкнуть мышью на кнопке  меню среды MS10), **снять** показания приборов и **занести** их в поля табл. 1.2 электронной тетради по работе. Методика определения угла сдвига фаз  $\varphi$  между напряжением и током описана в п. 4 раздела "Теоретические сведения ...".

**Убедиться** (см. рис. 1.3, а), что ток  $i_1$  в  $R_1L$ -ветви *отстаёт* по фазе от напряжения  $u$  на угол  $\varphi_1 = \arctg(X_L/R_1) = \arccos(P_1/UI_1)$ , где  $P_1$  – показание ваттметра. **Скопировать** и **занести** на страницу электронной тетради копию рисунка-чертёжа смоделированной схемы (см. рис. 1.5) с показаниями ваттметра и осциллограммами на экране осциллографа (см. рис. 1.3).

**Установить** частоту ЭДС  $f = 10$  кГц источника  $e(t)$  и с помощью переключателя **S** подключить  $R_2C$ -ветвь к источнику **e**. Показания приборов **занести** в табл. 14.

**Убедиться** (анализируя расположение осциллограмм на экране осциллографа), что ток  $i_2$  в  $R_2C$ -ветви *опережает* по фазе напряжение  $u$  на угол  $\varphi_2 = \arctg(-X_C/R_2) = -\arccos(P_2/UI_2)$ , где  $P_2$  – показание ваттметра.

**Рассчитать** полное  $Z = U/I$ , активное  $R = Z\cos\varphi$  и реактивное  $X = Z\sin\varphi$  сопротивления  $R_1L$ - и  $R_2C$ -ветви и занести их в поля табл. 1.2.

Так как индуктивное сопротивление катушки  $X_L = \omega L = 2\pi fL$ , Ом, а ёмкостное сопротивление конденсатора  $X_C = 1/\omega C = 1/2\pi fC$ , Ом, то:

– индуктивность катушки, включенной в  $R_1L$ -ветвь:

$$L = X_L/\omega = X_L/2\pi f, \text{ Гн или } L = 10^3 X_L/2\pi f, \text{ мГн};$$

– ёмкость конденсатора, включенного в  $R_2C$ -ветвь:

$$C = 1/(\omega X_C) = 1/(2\pi f X_C), \text{ Ф или } C = 10^6/(2\pi f X_C), \text{ мкФ}.$$

Таблица 14

| Ветвь  | Установлено |              | Измерено   |             |             |                     | Вычислено   |             |             |              |              |
|--------|-------------|--------------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|--------------|--------------|
|        | $E$ ,<br>В  | $f$ ,<br>кГц | $U$ ,<br>В | $I$ ,<br>мА | $P$ ,<br>Вт | $\varphi$ ,<br>град | $Z$ ,<br>Ом | $R$ ,<br>Ом | $X$ ,<br>Ом | $L$ ,<br>мГн | $C$ ,<br>мкФ |
| $R_1L$ |             | 1            |            |             |             |                     |             |             |             |              | -----        |
| $R_2C$ |             | 10           |            |             |             |                     |             |             |             | -----        |              |

Вычисленные значения индуктивности  $L$  катушки и ёмкости  $C$  конденсатора **занести** в табл. 13. **Сравнить** полученные значения  $R$ ,  $L$  и  $C$  с установленными их значениями в схеме цепи.

## Практическая работа № 7. Моделирование схем

**ЦЕЛЬ:** Изучить методы вычислений логических функций

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

В работе исследуется двухконтурная схема цепи (рис. 13) с двумя источниками постоянного напряжения, имитирующая, например, электрическую цепь электропитания автомобиля. В состав такой цепи входит: генератор с ЭДС  $E_1$  и внутренним сопротивлением  $R_{вт1}$ , аккумуляторная батарея с ЭДС  $E_2$  и внутренним сопротивлением  $R_{вт2}$  и эквивалентная нагрузка  $R$ .

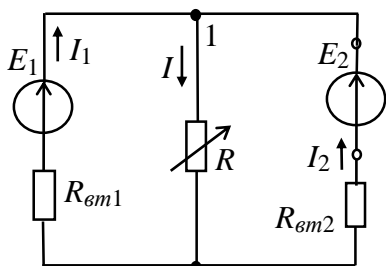


Рисунок 13

В соответствии с вариантом задания нужно установить значения ЭДС  $E_1$  и  $E_2$  источников энергии, их внутренние сопротивления  $R_{вт1}$  и  $R_{вт2}$  и сопротивление нагрузки  $R$ . При проведении опытов сопротивление потенциометра (нагрузки)  $R$  необходимо изменять в широких пределах. Условное положительное направление токов в ветвях схемы указано на рис. 13.

Расчет токов ветвей методом законов Кирхгофа

Для расчета токов ветвей используют следующие методы: метод законов Кирхгофа, метод наложения, метод контурных токов, метод двух узлов. Проведём расчёт токов в ветвях схемы (см. рис. 13) методом законов Кирхгофа. Согласно первому закону Кирхгофа (1ЗК) для узла 1:

$$I_1 + I_2 - I = 0.$$

Согласно второму закону Кирхгофа (2ЗК) для левого и правого контуров схемы имеем:

$$E_1 = R_{вт1}I_1 + RI; \quad E_2 = R_{вт2}I_2 + RI.$$

Используя формулы Крамера для системы уравнений

$$I_1 + I_2 - I = 0; \quad R_{вт1}I_1 + 0I_2 + RI = E_1; \quad 0I_1 + R_{вт2}I_2 + RI = E_2,$$

находим токи

$$I_1 = \frac{\Delta_1}{\Delta} = \frac{\begin{vmatrix} 0 & 1 & -1 \\ E_1 & 0 & R \\ E_2 & R_{см2} & R \end{vmatrix}}{\begin{vmatrix} 1 & 1 & -1 \\ R_{см1} & 0 & R \\ 0 & R_{см2} & R \end{vmatrix}}; \quad I_2 = \frac{\Delta_2}{\Delta} = \frac{\begin{vmatrix} 1 & 0 & -1 \\ R_{см1} & E_1 & R \\ 0 & E_2 & R \end{vmatrix}}{\begin{vmatrix} 1 & 1 & -1 \\ R_{см1} & 0 & R \\ 0 & R_{см2} & R \end{vmatrix}}; \quad I = I_1 + I_2.$$

Мощность, потребляемая нагрузкой  $R$ , равна (в Вт)

$$P = RI^2.$$

Составим баланс мощностей схемы цепи. Алгебраическая сумма мощностей, развиваемых источниками энергии, равна сумме мощностей, потребляемой нагрузкой и теряемой внутри источников, т. е.

$$\pm E_1 I_1 \pm E_2 I_2 = RI^2 + R_{см1} I_1^2 + R_{см2} I_2^2$$

(знак "плюс" в произведениях  $E_k I_k$  ставится при совпадении выбранных направлений ЭДС  $E_k$  и тока  $I_k$  источника, а знак "минус" – при противоположном их направлении).

**Потенциальная диаграмма контура**

Потенциальная диаграмма – это распределение потенциалов узлов схемы в зависимости от сопротивлений элементов контура, т. е.  $\varphi = f(R)$ .

Для построения потенциальной диаграммы, например, для наружного контура (см. рис. 14), необходимо знать величину и направление токов ветвей и ЭДС источников.



Рисунок 14

Пусть  $E_1 = 14$  В;  $R_{вт1} = 0,5$  Ом;  $E_2 = 12$  В;  $R_{вт2} = 2$  Ом;  $R = 10$  Ом.

Воспользовавшись калькулятором ElCalc, находим согласно (3.1) токи ветвей (рис. 3.2):

$$I_1 \approx 1,85 \text{ А}; I_2 \approx -0,54 \text{ А};$$

$$I \approx 1,31 \text{ А}.$$

Заземлим точку а (рис. 3.3, а), т. е. примем её потенциал  $\varphi_a = 0$ .

Так как ток  $I_1$  протекает от точки а к точке b, то потенциал точки b

$$\varphi_b = \varphi_a - R_{вм1} I_1 = 0 - 0,5 \cdot 1,85 = -0,925 \text{ В},$$

а потенциал точки с больше потенциала  $\varphi_b$  на 14 В, т. е.

$$\varphi_c = \varphi_b + E_1 = -0,925 + 14 = 13,075 \text{ В}$$

(на диаграмме рис. 3.3, б скачок потенциала на 14 В за счет сторонних сил).

Потенциал точки d:  $\varphi_d = \varphi_c - E_2 = 13,075 - 12 = 1,075 \text{ В}$ , а потенциал

точки а:  $\varphi_a = \varphi_d - R_{вм2} I_2 = 1,075 - 2 \cdot 0,54 \approx 0 \text{ В}$ .

Возвращение в исходную точку а с нулевым потенциалом после расчёта потенциалов всех точек вдоль выбранного контура подтверждает правильность расчёта цепи.

### ХОД РАБОТЫ

**Задание 1.** Рассчитать токи ветвей схемы (см. рис. 13) методом законов Кирхгофа, напряжение  $U_1 = E_1 - R_{вт1} I_1$  и мощность, потребляемую нагрузкой R. Проверить результаты расчета методом баланса мощностей и построением потенциальной диаграммы наружного контура.

При расчетах использовать параметры элементов схемы, определяемые согласно выражениям:

$$E_1 = 5 + N, \text{ В}; R_{вт1} = 0,5 + \text{int}(N/8), \text{ Ом}; R = 10 + N, \text{ Ом};$$

$$E_2 = 3 + N, \text{ В}; R_{вт2} = 2 + \text{int}(N/5), \text{ Ом},$$

где N – номер записи фамилии студента в учебном журнале группы.

Результаты расчёта токов, напряжения  $U_1$  и мощности P при сопротивлении нагрузки  $R = 10 + N$ , Ом занести в правый столбец табл. 15.

Таблица 15

| Измерено при сопротивлении нагрузки R [Ом], равном: |                  |     |    |    |   |      |      |      |           | Рас-<br>счи-<br>тано<br>при R |
|---|------------------|-----|----|----|---|------|------|------|-----------|-------------------------------|
| Токи ветвей,<br>напряжение<br>U1 и мощ-<br>ность P  | $\infty$<br>(XX) | 10R | 5R | 2R | R | 3/4R | 1/2R | 1/4R | 0<br>(K3) |                               |
| I1, А   |                  |     |    |    |   |      |      |      |           |                               |

|       |  |  |  |  |  |  |  |  |  |
|-------|--|--|--|--|--|--|--|--|--|
| I2, A |  |  |  |  |  |  |  |  |  |
| I3, A |  |  |  |  |  |  |  |  |  |
| U1, В |  |  |  |  |  |  |  |  |  |
| P, Вт |  |  |  |  |  |  |  |  |  |

**Задание 2.** Собрать на рабочем поле среды MS схему для испытания двухконтурной схемы цепи постоянного тока (рис. 15).

Установить в диалоговых окнах компонентов их параметры или режимы работы:

- значения параметров элементов схемы цепи;
- режим DC работы приборов; внутренние сопротивления амперметра RA = 1 нОм, а вольтметров RV = 10 МОм.

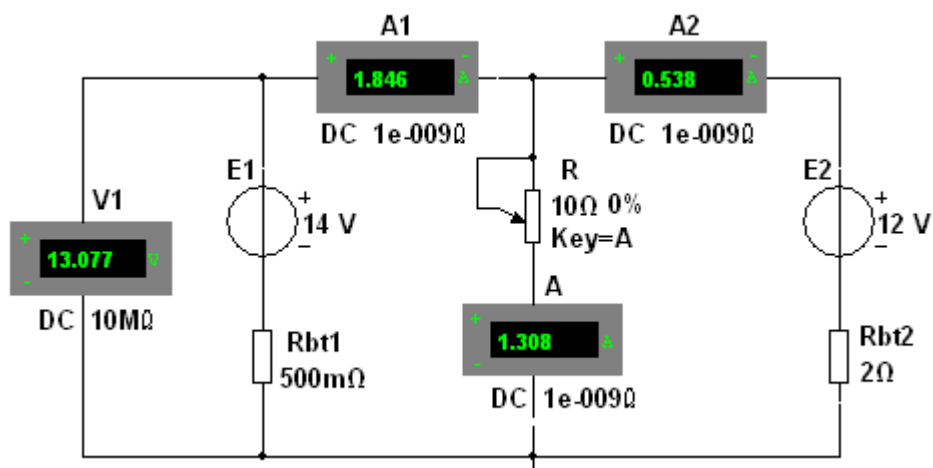



Рисунок 15

Запустить программу (щелкнув мышью на цифре 1 кнопки ) и записать показания приборов в столбец табл. 3.1 при сопротивлении нагрузки R = 10 + N, Ом.

**Задание 3.** Снять и построить внешнюю характеристику U1(I1) источника напряжения E1, т. е.

$$U1 = E1 - R_{bt1}I1.$$

С этой целью необходимо ступенчато изменять (посредством нажатия на клавишу A клавиатуры) сопротивление потенциометра (нагрузки) R от режима холостого хода (ХХ) до режима короткого замыкания (КЗ), а показания приборов заносить в табл. 14.

По результатам измерений построить (в масштабе) внешнюю характеристику U1(I1) источника E1.

**Задание 4.** Рассчитать и построить график зависимости мощности P от сопротивления R, т. е. P(R). Отметить на графике координаты максимальной мощности.

## Практическая работа № 8. Моделирование логических схем

**ЦЕЛЬ:** изучить способы моделирования логических схем

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

Логический элемент представляет собой устройство, предназначенное для выполнения различных логических операций над дискретными сигналами при двоичном способе их представления



Таблица истинности — это таблица, описывающая логическую функцию. Под «логической функцией» в данном случае понимается функция, у которой значения переменных (параметров функции) и значение самой функции выражают логическую истинность. Например, в двузначной логике они могут принимать значения «истина» либо «ложь».

Табличное задание функций встречается не только в логике, но для логических функций таблицы оказались особенно удобными рис. 16.

|                           |                          |                               |
|---------------------------|--------------------------|-------------------------------|
| <b>Конъюнкция</b>         | <b>Дизъюнкция</b>        | <b>Сложение по модулю</b>     |
| $a \ b \ a \wedge b$      | $a \ b \ a \vee b$       | $a \ b \ a \oplus b$          |
| 0 0 0                     | 0 0 0                    | 0 0 0                         |
| 0 1 0                     | 0 1 1                    | 0 1 1                         |
| 1 0 0                     | 1 0 1                    | 1 0 1                         |
| 1 1 1                     | 1 1 1                    | 1 1 0                         |
| <b>Импликация</b>         |                          | <b>Эквиваленция</b>           |
| $a \ b \ a \rightarrow b$ |                          | $a \ b \ a \leftrightarrow b$ |
| 0 0 1                     |                          | 0 0 1                         |
| 0 1 1                     |                          | 0 1 0                         |
| 1 0 0                     |                          | 1 0 0                         |
| 1 1 1                     |                          | 1 1 1                         |
| <b>Штрих Шеффера</b>      | <b>Стрелка Пирса</b>     | <b>Отрицание</b>              |
| $a \ b \ a   b$           | $a \ b \ a \downarrow b$ | $a \ \neg a$                  |
| 0 0 1                     | 0 0 1                    | 0 1                           |
| 0 1 1                     | 0 1 0                    | 1 0                           |
| 1 0 1                     | 1 0 0                    |                               |
| 1 1 0                     | 1 1 0                    |                               |

Рисунок 16

Графическое отображение логических элементов представлено на рис.17.

Набор простейших логических функций, позволяющих реализовать любые другие функции называются логическим базисом.

Функции И, ИЛИ, НЕ не являются минимальным логическим базисом, так как сами могут быть представлены через другие функции

| Название  | И           | ИЛИ     | И-НЕ (NAND)                                    | ИЛИ-НЕ (NOR)                                | XOR (исключающее ИЛИ)  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|-------------|---------|--|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F   | $x \cdot y$ | $x + y$ | $\overline{x \cdot y}$                         | $\overline{x + y}$                          | $x \oplus y = x \cdot \overline{y} + \overline{x} \cdot y = x \leftrightarrow y$ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Графическое обозначение   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <table border="1"> <tr><th>x</th><th>y</th></tr> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td></tr> </table> | x           | y       | 1  | 1   | 1  | 0 | 0 | 1 | 0 | 0 | <table border="1"> <tr><td>1</td></tr> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>0</td></tr> </table> | 1 | 0 | 0 | 0 | <table border="1"> <tr><td>1</td></tr> <tr><td>1</td></tr> <tr><td>1</td></tr> <tr><td>0</td></tr> </table> | 1 | 1 | 1 | 0 | <table border="1"> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>1</td></tr> <tr><td>1</td></tr> </table> | 0 | 1 | 1 | 1 | <table border="1"> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>0</td></tr> </table> | 0 | 0 | 1 | 0 | <table border="1"> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>1</td></tr> <tr><td>0</td></tr> </table> | 0 | 1 | 1 | 0 |
| x   | y           |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1           |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 0           |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1           |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0           |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   |             |         |  |   |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |             |         | Высказывания x и y не могут быть оба истинными | Выдаёт истину, когда оба её аргумента ложны | Выдаёт истину, когда одно из входных значений истина                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |             |         | Базис Шеффера                                  | Базис Пирса                                 |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Рисунок 17

## ХОД РАБОТЫ

**Задание 1.** Построение схемы логического цифрового устройства в программе

1.С помощью логических элементов 2 И-НЕ соберём цифровое устройство, создающий прямоугольные импульсы требуемой частоты. Чтобы измерить частоту импульсов воспользуемся частотомером (frequency counter), а проверим его показания с помощью осциллографа рис. 18.

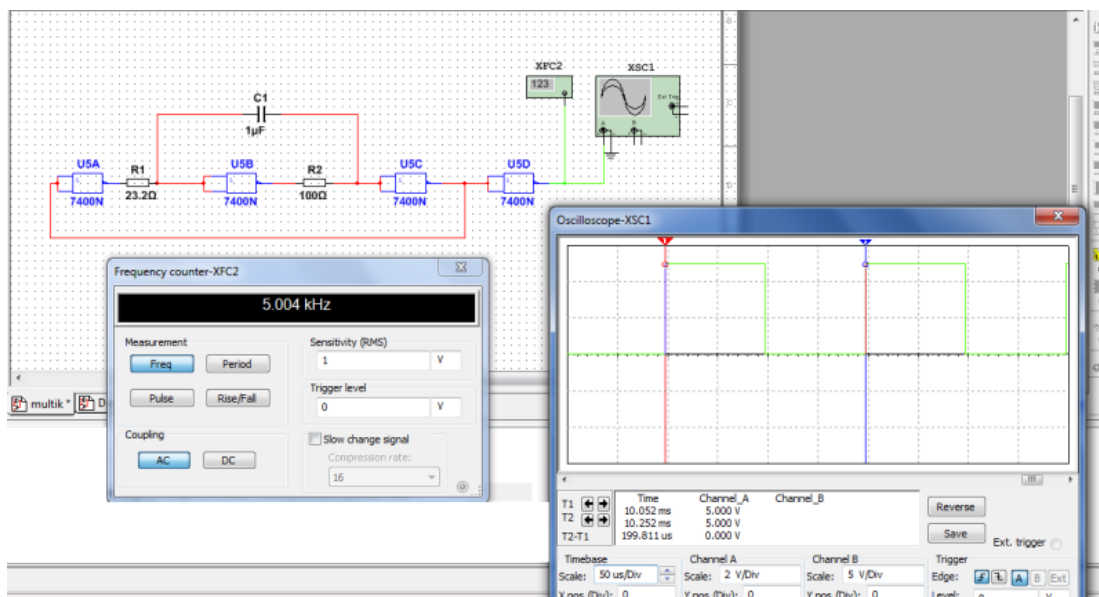


Рисунок 18

2. Задать частоту 5 кГц;
3. Подобрать опытным путём требуемые значения конденсатора и резисторов.
4. Запустить схему и проверить, что частотомер показывает приблизительно 5 кГц.
5. На осциллограмме отмечаем период импульса, который в нашем случае равен 199,8 мкс.

6. Тогда частота равна

$$f = \frac{1}{T} = \frac{1}{199.8 \cdot 10^{-6}} \approx 5 \text{ кГц}$$

7. Создать новый проект
8. С панели «Функций» (справой стороны) выбрать значок «Логической функции НЕ» это элемент битовой логики, обозначающий отрицание или инверсия, мышкой перетащить на рабочую область.
9. Проанализируйте работу реле: определите, как в данном случае работает логическая операция «НЕ», заполните таблицу данными. Попеременно, а потом одновременно подайте сигнал на вход (1, 2,3) и определите результат на выходе табл. 16.

Таблица 16

| Вход I | and | not | Выход Q |
|--------|-----|-----|---------|
| I1     | 1   |     |         |
| I2     | 0   |     |         |
| I3     | 1   |     |         |

10. Добавить еще один элемент «И» «НЕ», чтобы расширить количество входов: для этого необходимо удалить соединение со вторым входом (щелкнуть левой кнопкой мыши по соединительной линии и нажать кнопку Delete).
11. Перетащить на рабочую область логические элементы «И», «НЕ» и осуществить соединение/
12. Проанализируйте данную ситуацию и определите, в каком случае в двух выходах будет сигнал ответ оформите в виде таблицы табл. 17, заполните ее данными.

Таблица 17

| Вход I |   | and | nod | Выход Q1 |  | and | nod | Выход Q2 |  |
|--------|---|-----|-----|----------|--|-----|-----|----------|--|
| I1     | 0 |     |     |          |  |     |     |          |  |
| I2     | 1 |     |     |          |  |     |     |          |  |
| I3     | 1 |     |     |          |  |     |     |          |  |

### Практическая работа № 9. Моделирование схемы электроснабжения квартиры (часть 1)

**ЦЕЛЬ:** создать модель схемы электроснабжения квартиры

#### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ.

В зависимости от строительных конструкций жилых зданий применяют две типовые схемы скрытых квартирных проводок: раздельную и радиальную. По раздельной схеме выполняют групповые сети, когда питание штепсельных розеток и светильников осуществляют раздельными группами. В этом случае при монтаже электропроводки провода группы штепсельных розеток прокладывают по полу квартиры и по стене с подъемом на некоторое расстояние от пола, необходимое для установок штепсельных розеток; провода группы светильников прокладывают по перекрытию вышележащего этажа.

По радиальной схеме выполняют групповые сети, когда от квартирного щитка провода прокладывают отдельными группами по перекрытию вышележащего этажа по наикратчайшим расстояниям к штепсельным розеткам и светильникам. При радиальной схеме от каждой группы осуществляется смешанное питание светильников и штепсельных розеток.

Групповую сеть в квартирах выполняют однофазными линиями — двумя самостоятельными группами, которые присоединяют к квартирному щитку, установленному на лестничной клетке.

Электрические сети жилого дома разделяют на питающие (магистральные) и групповые (распределительные). К питающим относятся линии от вводно-распределительного устройства (главного распределительного щита) до этажных групповых щитков.

В многоэтажных зданиях питающие линии представляют собой систему вертикальных линий «стояков» с подводкой питания к ним от главного распределительного устройства.

К групповым (распределительным) относятся линии от групповых этажных щитков к осветительным токоприемникам в квартирах.

Групповые щитки на этажах здания располагают так, что оказывается целесообразным разветвить питающую линию «стояк» на несколько направлений в квартиры уже по групповым линиям. Осветительная групповая линия квартиры начинается от группового этажного щитка, на котором сосредоточены аппараты защиты, учета, выключатели. В многоэтажных зданиях групповые щитки обычно располагают на лестничных клетках или вблизи них.

В число светильников, которые обслуживают групповые линии, входят и контактные разъемные соединения. Рекомендуется при большом числе разъемных соединений питать их отдельными группами от щитка. Отдельные группы выделяют также для лестничных клеток и проходов.

Электрические схемы квартирных электропроводок разнообразны, но в каждой из них применяют несколько одинаковых элементов: лампы накаливания, люминесцентные лампы, люстры, штепсельные розетки для включения бытовых электроприборов, настольные лампы, звонки, защитные аппараты (автоматы, плавкие предохранители и т. д.), счетчики энергии.

### ХОД РАБОТЫ

#### Задание.

1. По данному чертежу проекта составить монтажную и многолинейную схемы соединения проводов в ответвительных коробках и осуществить сборку схемы (рис.19).

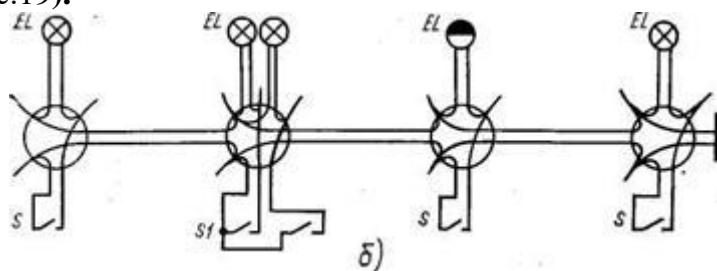


Рисунок 19

2. Ознакомиться с устройством установочных аппаратов и применяемыми для электропроводки проводами АППВ и АППВС.

3. На основании этажного плана составить монтажную схему электропроводки однокомнатной квартиры (рис. 20).

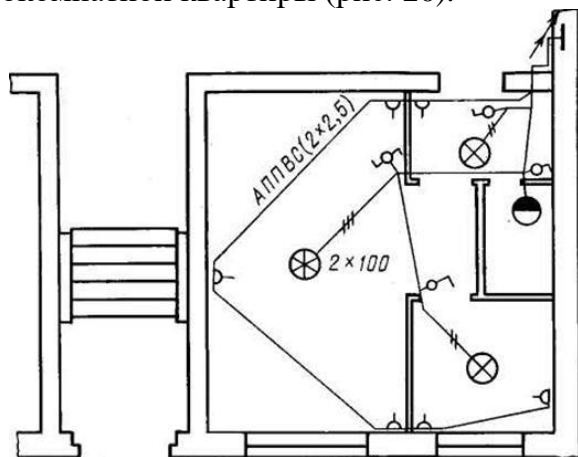


Рисунок 20

4. По монтажной составить многолинейную схему соединения проводов в ответвительных коробках (рис. 19).

### Практическая работа № 10. Моделирование схемы электроснабжения квартиры (часть 2)

ЦЕЛЬ: создать модель схемы электроснабжения квартиры

#### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

В зависимости от строительных конструкций жилых зданий применяют две типовые схемы скрытых квартирных проводок: раздельную и радиальную. По раздельной схеме выполняют групповые сети, когда питание штепсельных розеток и светильников осуществляют раздельными группами. В этом случае при монтаже электропроводки провода группы штепсельных розеток прокладывают по полу квартиры и по стене с подъемом на некоторое расстояние от пола, необходимое для

установок штепсельных розеток; провода группы светильников прокладывают по перекрытию вышележащего этажа.

По радиальной схеме выполняют групповые сети, когда от квартирного щитка провода прокладывают отдельными группами по перекрытию вышележащего этажа по наикратчайшим расстояниям к штепсельным розеткам и светильникам. При радиальной схеме от каждой группы осуществляется смешанное питание светильников и штепсельных розеток.

Групповую сеть в квартирах выполняют однофазными линиями — двумя самостоятельными группами, которые присоединяют к квартирному щитку, установленному на лестничной клетке.

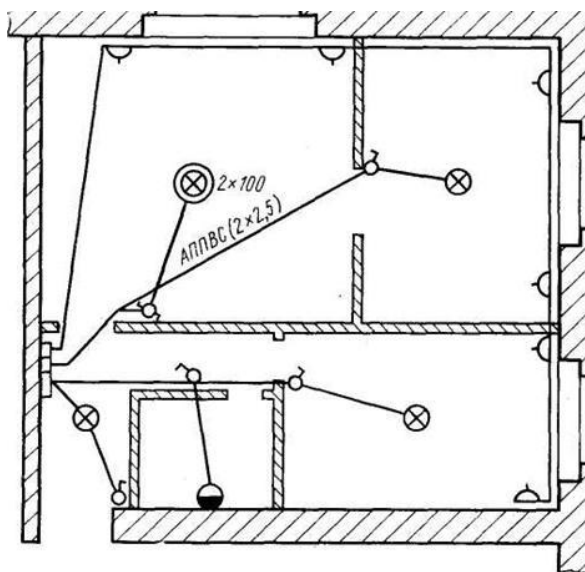


Рисунок 21

**Задание:** Создать схему по этажному плану электропроводки двухкомнатной квартиры рис.21..

2. Составить многолинейную схему розеточной группы рис.22

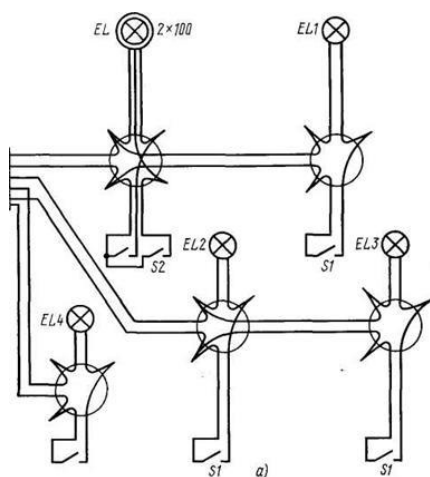


Рисунок 22

### Практическая работа №11. Запись математических выражений и вычисление их значений при заданных исходных данных

**ЦЕЛЬ:** изучить правила записи математических выражений и вычислений

## ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

В Mathcad, как и в языках программирования, константы, переменные и функции могут иметь имя-идентификатор, которое может состоять из следующих элементов:


- прописных и строчных латинских и/или греческих букв;
- цифр от 0 до 9;
- символов подчеркивания ( ), штриха (′), процента (%), бесконечности ( $\infty$ ).

В системе Mathcad имеется ряд ограничений на использование некоторых символов в начале имени, поэтому обычно имена начинаются с буквы. Все символы имени должны быть напечатаны одинаковым шрифтом.

При составлении имени следует учитывать, что в отличие от большинства других программ Mathcad различает в формулах не только прописные и строчные буквы, но даже стиль их оформления. Например, переменные  $N$ ,  $N$ ,  $n$ ,  $n$  рассматриваются как различные и не взаимозаменяемы [2].

Примеры правильных имен-идентификаторов Mathcad:

$S$ ,  $ht_$ ,  $x^2$ ,  $w_{min}$ ,  $DORk\%$ ,  $pw$ ,  $\lambda_5$ ,  $\Delta f$ .


Ввод имени переменной с нижним индексом, например,  $w_{min}$ , осуществляется следующим образом: вводится часть имени без индекса ( $w$ ), затем на клавиатуре набирается десятичная точка (на экране она не отображается), после чего вводятся символы нижнего индекса ( $min$ ). Не следует путать буквенный нижний индекс, являющийся частью имени переменной, с нижним индексом элемента массива данных. Выглядят они одинаково, но по смыслу различны. Нижний индекс массива осуществляет ссылку на элемент массива и вводится с помощью кнопки  палитры векторных и матричных операций.

Mathcad не делает различий между именами переменных и именами функций. Если вначале была определена функция  $f(x)$ , а затем задана переменная  $f$ , то после этого использование определения  $f(x)$  окажется невозможным.

### Ввод математических выражений

Важным преимуществом Mathcad является то, что он отображает формулы практически в том же виде, как их обычно пишут на бумаге. Ввод символов, имеющих на клавиатуре, можно производить непосредственно с клавиатуры или с помощью соответствующих кнопок палитр символов. В качестве разделителя между целой и дробной частями десятичных чисел используется точка. При вводе произведения знак умножения между сомножителями обязателен (при этом Mathcad преобразует оператор умножения «\*» в традиционную точку). Общепринятый оператор деления «/» Mathcad интерпретирует как горизонтальную дробную черту стандартного вида.


При вводе составных операторов в области формулы автоматически появляются слоты – заполнители в виде черных прямоугольников, предназначенные для записи соответствующих операндов, например, для ввода корня произвольной степени они имеют вид:


. Перемещение по слотам осуществляется щелчком левой кнопки мыши на них или клавишей Tab.

Содержимое уголкового курсора становится первым операндом очередного вводимого оператора, т. е. каждое следующее действие при вводе формулы относится к данным, заключенным в уголкового курсора. Это свойство позволяет значительно упростить процесс ввода сложных формул, где порядок операций определяется скобками. Элементы формулы, которые в рамках операции должны рассматриваться как единое целое, нужно «охватить» уголковым курсором, а затем выпол-

нить относящееся к ним действие. Управление уголковым курсором осуществляется следующим образом: нажатие на клавиатуре клавиши  $\uparrow$  или пробела увеличивает объем содержимого уголкового курсора (т. е. объединяет операторы), нажатие клавиши  $\downarrow$  – уменьшает.


При правильном вводе формулы Mathcad автоматически заключает в скобки нужные ее фрагменты. При необходимости скобки можно удалять и вставлять по одной обычными способами редактирования. Чтобы заключить содержимое уголкового курсора в скобки, следует нажать клавишу ' (апостроф). Для удаления парных скобок нужно поместить в уголкового курсор фрагмент формулы, включая скобки, и нажать клавиши Delete или BackSpace. Того же результата можно добиться, поместив курсор ввода справа от левой скобки и нажав BackSpace.

Для ввода стандартных математических функций используется кнопка  на панели инструментов. Наиболее часто употребляемые функции имеются и на ариф-

метической палитре . Кроме того, имя функции можно вводить побуквенно непосредственно с клавиатуры. Аргумент функции всегда записывается в скобках (табл. 1). По умолчанию аргумент любой тригонометрической функции предполагается в радианах. При необходимости можно ввести единицы измерения конкретной переменной или константы. Чтобы это сделать, нужно просто умножить ее на соответствующую единицу, которая выбирается из списка в меню Математика  $\rightarrow$  Единицы  $\rightarrow$  Вставить единицы, например,  $\sin 90^\circ$  следует ввести как  $\sin(90 \cdot \text{deg})$ .


Степенное выражение вида  $\sin 2x$  в Mathcad записывается как  $\sin(x)2$ .


Ввод операторов суммирования ( $\Sigma$ ), произведения ( $\Pi$ ), интегрирования ( $\int$ ) и некоторых других осуществляется аналогично стандартным функциям с помощью

соответствующих кнопок палитры операторов математического анализа .

В Mathcad для разных целей используются три разных знака равенства.

Значения переменных и функций задаются при помощи оператора присваивания  $:=$ , для ввода которого можно нажать на клавиатуре клавишу двоеточия  $:$  или

использовать кнопку  палитры простых операторов. Например:  $x:=1$ ,  $y:=2 \cdot \sin(x+5)$ ,  $M:=4 \cdot \pi$ .

Знак булева равенства используется в логических операциях, в том числе при записи уравнений. Для его ввода на клавиатуре можно нажать CTRL+ = или использовать кнопку  палитры простых операторов. Например:  $x + y = 5 \cdot x$ .

Обычный знак равенства используется для получения результата, например:  $x =$  или  $\sin(2 \cdot \pi + 7) =$ . В ответ Mathcad выводит текущее значение записанной слева от знака равенства переменной или результат расчета по заданной формуле.

Рассмотрим основные правила построения математических выражений и выполнения расчетов в Mathcad на конкретном примере.

$$y = \frac{2 \sin x^2}{\sqrt{x+a}} - 5$$

Пусть нужно вычислить  $y$  при  $a = 0,8$ ;  $x = 4,56$ .

В табл. 2 приведена последовательность действий для решения этой задачи, результат работы см. на рис. 4.

Пропущенные при вводе формулы символы можно вводить обычным образом, установив курсор в нужное место. Для вставки пропущенного операнда после име-

ющегося операнда нужно поместить в уголкоый курсор имеющийся операнд, сначала ввести знак математической операции, а затем набрать пропущенный операнд. Вставка пропущенного операнда перед имеющимся операндом выполняется аналогично, за исключением того, что после заключения операнда в уголкоый курсор нужно перейти в режим вставки символов, нажав клавишу Insert (при этом уголкоый курсор изменит цвет с синего на красный, а срезанный угол будет слева).

Пример ввода формулы и вычисления в Mathcad табл 18

Таблица 18

| Последовательность действий   | Результат   |
|---|---|
| Выводим на экран арифметическую палитру символов  | см. рис. 2  |
| Вводим исходные данные: $a := 0.8$ ; $x := 4.56$  | $a := 0.8$ $x := 4.56$                            |
| Вводим $y$ , затем $-$  | $y :=$  |
| Вводим $2$ и знак умножения $*$   | $y := 2 \cdot$                                    |
| На арифметической палитре выбираем $\sin$<br>(или из списка функций по кнопке $f(x)$ выбираем $\sin(z)$ ) | $y := 2 \cdot \sin$                               |
| Вводим $x$  | $y := 2 \cdot \sin(x)$                            |
| На арифметической палитре выбираем $x^y$<br>(или на клавиатуре нажимаем $1^1$ )                           | $y := 2 \cdot \sin(x^)$                           |
| Вводим степень $2$ , затем четыре раза нажимаем $1^1$ для охвата всего числителя уголкоый курсором        | $y := 2 \cdot \sin(x^2)$                          |
| Вводим символ деления $/$   | $y := \frac{2 \cdot \sin(x^2)}{}$                 |
| На арифметической палитре выбираем символ $\sqrt{\quad}$  | $y := \frac{2 \cdot \sin(x^2)}{\sqrt{\quad}}$     |
| Вводим $x + a$ , затем четыре раза нажимаем $1^1$ для охвата уголкоый курсором всей дроби                 | $y := \frac{2 \cdot \sin(x^2)}{\sqrt{x + a}}$     |
| Вводим с клавиатуры $-$ и $5$   | $y := \frac{2 \cdot \sin(x^2)}{\sqrt{x + a}} - 5$ |
| На свободном месте листа для получения результата расчета набираем $y = 1$                                | $y = -4.196$<br>(см. рис. 4)                      |

### Особенности вычислений в Mathcad

В ходе расчетов формулы обрабатываются последовательно слева направо и сверху вниз. Если расположить значение переменной ниже или правее формулы, в которую она входит, документ становится логически непоследовательным, что приводит к невозможности выполнения вычислений. Переменная в формуле является неопределенной и выделяется черным цветом (рис. 23).



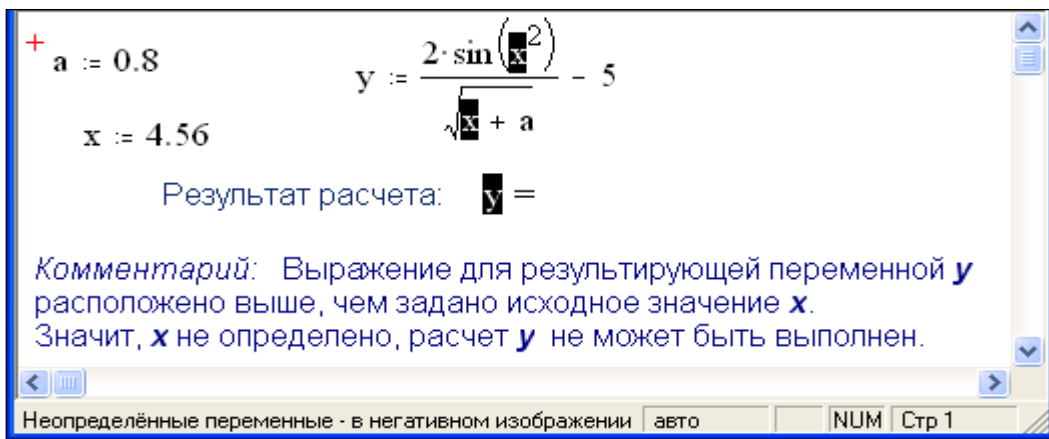


Рисунок 23

Для исправления ошибки нужно просто выделить и переместить любую из областей определения –  $x$  или  $y$ , расположив их в логической последовательности (см. рис. 4).

Если на рабочем листе Mathcad какая-либо переменная была определена несколько раз, то в каждом конкретном месте документа в расчетах используется то ее значение, которое обработано Mathcad последним перед этим местом, а предыдущие значения игнорируются (рис. 24).

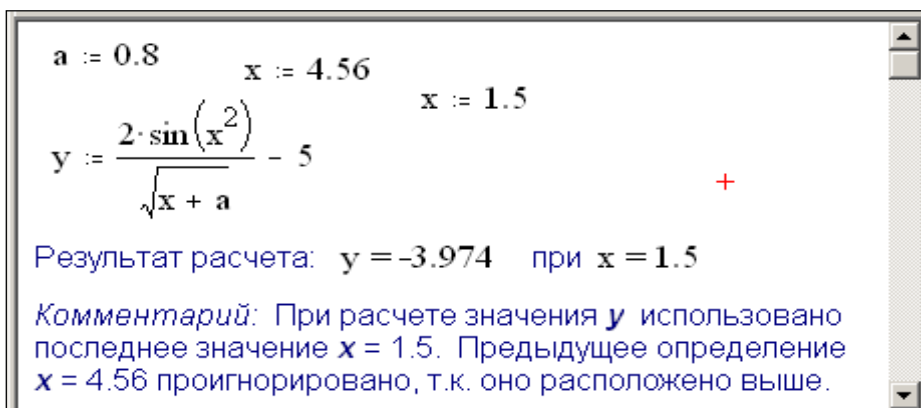


Рисунок 24

Если не установлено иное, Mathcad выводит на экран результаты вычислений с точностью до 0,001. Изменить количество выводимых десятичных знаков можно в меню Математика → Формат числа → Выводимая точность. Данная установка не влияет на точность вычислений. Она затрагивает только вывод результата на экран и определяет количество цифр, отображаемых после десятичной точки. Например, число  $\pi$  при выводимой точности, равной 1, будет отображаться как 3.1, а при точности, равной 9, – как 3.141592654.

Если выводимую точность требуется изменить только для конкретного значения, то перед выполнением указанной выше операции следует поместить это значение в уголкоый курсор.

Независимо от заданной точности вывода результата в системе Mathcad всегда поддерживается максимальная точность вычислений. При решении уравнений, вычислении интегралов и производных итерационными (приближенными) методами точность вычислений (погрешность численных расчетов) по умолчанию составляет 0,001. Чтобы установить другую точность вычислений, нужно задать требуемое значение для встроенной переменной TOL в меню Математика → Встроенные переменные → TOL или ввести его непосредственно в тексте документа, например,

TOL:=0.000001, перед соответствующей формулой или в самом начале рабочего листа.

Прервать процесс вычислений можно, нажав клавишу Esc. Mathcad отметит выражение, вычисление которого было прервано, сообщением об ошибке «прервано». Для возобновления вычислений нужно нажать F9.

### Практическая работа №12. Запись математических выражений и вычисление их значений при заданных исходных данных

**ЦЕЛЬ:** отработать приемы записи математических выражений и вычислений:

#### ХОД РАБОТЫ

**Задание 1.** Выполнить работу по образцу, представленному на рис. 25.

**Задание 2.** Вычислить значения функций при заданных значениях переменных в соответствии с табл.19 и табл. 20. При вычислении интеграла переменную интегрирования определить самостоятельно. Сравнить полученные результаты с табличными.

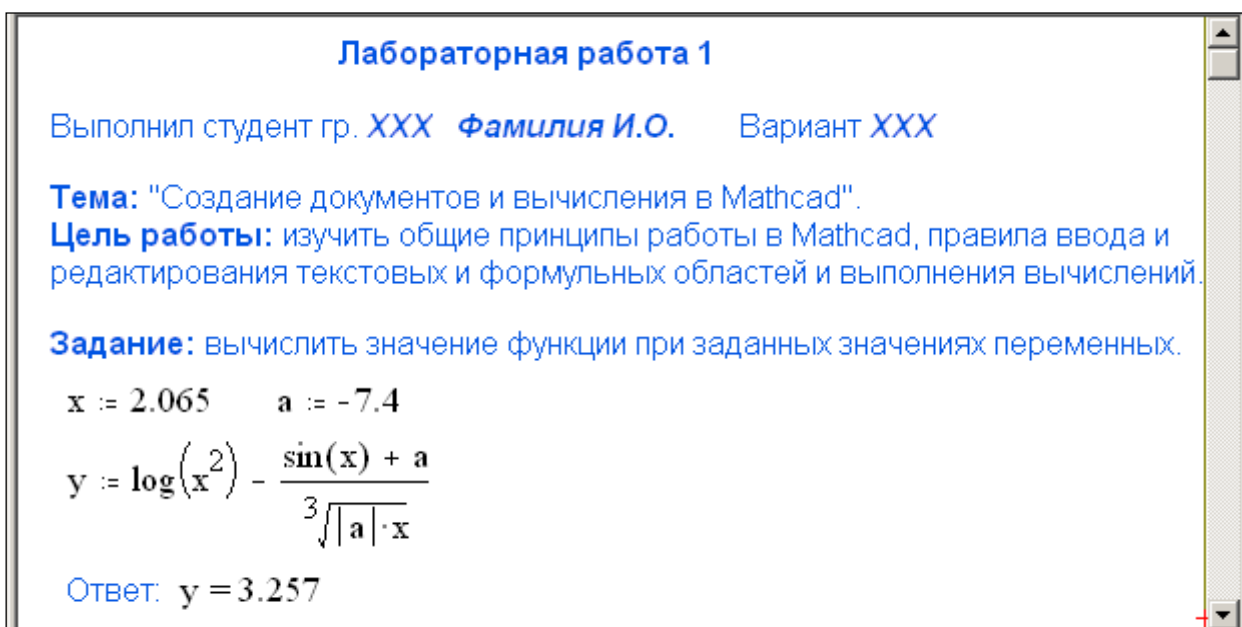


Рисунок 25

#### Арифметические выражения

Таблица 19

| Вариант | Исходные данные  | Функция  | Результат   |
|---------|------------------|--|-------------|
| 1       | 2                | 3  | 4           |
|         | $x = 5; a = 3,3$ | $y = \sqrt[3]{\frac{0,5 \cos^2 x^a + e^{x-3}}{2 \ln x^2}}$                   | $y = 0,539$ |
|         | $x = 5; a = 3,3$ | $y = \frac{e^{\sin x} + \sqrt[3]{\cos x}}{\ln(\sqrt{a^2 + 3,71 - a}) + 4,5}$ | $y = 0,27$  |

|                    |   |              |
|--------------------|---|--------------|
| $x = 5; a = 3,3$   | $y = \sqrt[4]{\frac{7,61 \cos^3(3,5x) + 0,6}{ a^3 - \sin^3 \frac{1}{x} }}$      | $y = 0,371$  |
| $x = 0,5; a = 3,3$ | $y = \frac{12,7 \arcsin^3(3x^2)}{\ln^2 \sqrt{x + e^a}}$                         | $y = 2,814$  |
| $x = 5; a = 3,3$   | $y = 2,31 e^{x \cos(ax)} - \sqrt{ \operatorname{tg}(3x) - 12,7 ^3}$             | $y = -46,72$ |
| $x = 0,5; a = 3,3$ | $y = 2,17 a^3 \sqrt{e^{0,8x^3} - \cos^5 \frac{1}{x}}$                           | $y = 82,443$ |
| $x = 5; a = 3,3$   | $y = \sqrt{\cos^3 85^\circ + \sqrt{1,24 + e^{x^{-2a}}}}$                        | $y = 1,22$   |
| $x = 5$            | $y = \lg \sqrt{\frac{ \sin x }{x + 2 \operatorname{arctg} \ln^2 x}}$            | $y = -0,444$ |
| $x = 0,5$          | $y = \arccos^2 \sqrt{0,71x + 2} -  \sin^2 \cos \sqrt{x} $                       | $y = -1,46$  |
| $x = 0,5$          | $y = 2,5 \arcsin^2(\cos 64^\circ + e^{-3x})$                                    | $y = 1,306$  |
| $x = 0,5$          | $y = \arccos^2 x - \sqrt{3x - \left  \cos \frac{x}{2 \ln^5 x} \right }$         | $y = -0,125$ |
| $x = 0,5; a = 3,3$ | $y = \frac{\arcsin^3(e^{x-a}) + 1,6}{\sqrt{ x - 5a }}$                          | $y = 0,4$    |
| $x = 0,5; a = 3,3$ | $y = \sqrt{\frac{\sqrt[3]{\lg(5,3x) + x/a}}{\operatorname{arctg}(\sin^5(3x))}}$ | $y = 1,076$  |
| $x = 0,5; a = 3,3$ | $y = \frac{\sqrt{x^3 + \sin^3(x+4)^2} - \ln 5,7}{\sqrt[3]{x + a^5}}$            | $y = -0,096$ |
| $x = 0,5$          | $y = \frac{\operatorname{arctg}^2(\sin \sqrt{1 + \pi x})}{\cos 12^\circ}$       | $y = 0,63$   |
| $x = 0,5; a = 3,3$ | $y = \frac{\sqrt{x^2 + 1} \sqrt[3]{ x - a } \sin^2 x^3}{1 + e^{x^2}}$           | $y = 0,011$  |

Выражения с использованием операторов математического анализа

Таблица 20

| Исходные данные | Функция   | Результат  | Функция                                      | Результат |
|-----------------|---|------------|--|-----------|
| 2               | 3   | 4          | 5  | 6         |
| $x = 15$        | $b = \sum_{k=3}^9 \operatorname{tg}(x) + \prod_{n=4}^6 \frac{n^3}{x^2} \sin(n^3)$ | $b = -6,1$ | $\int_{-2}^{12} \frac{\cos(2x)}{\sqrt{x+5}}$ | $-0,34$   |

|              |  |              |  |         |
|--------------|--|--------------|--|---------|
| $a = 1,5$    | $m = \left  \prod_{k=1}^8 \left( a^{-6} + \sum_{u=2}^7 \operatorname{tg}(5+a) \right) \right $       | $m = 15,57$  | $\int_5^7 \ln^3 t - t^2$               | -61,151 |
| $a = 5,8$    | $p = \prod_{m=1}^3 (am + \sqrt[3]{m^2}) + \sum_{n=1}^3 \frac{5}{n^2}$                                | $p = 1754$   | $\int_1^{11} \frac{\sin(z+5)}{z^2}$    | 0,232   |
| $u = 0,5$    | $t = \sum_{q=29}^{43} \prod_{p=3}^9 \frac{\cos(u - (25u)^2)}{\sin(25u - u^2)}$                       | $t = -2,36$  | $\int_2^4 \sqrt{x} + e^x$              | 50,657  |
| $x = 0,99$   | $p = \sum_{m=-4}^1 \frac{\prod_{k=1}^3 m(k^2 x + kx^2)}{20x}$  | $p = -691,1$ | $\int_0^9 \lg s - \sqrt{s^3}$          | -92,524 |
| $x = 0,57$   | $t = \sum_{v=1}^2 \prod_{i=2}^3 \frac{\operatorname{tg}(5\sqrt{v})}{5 \cos(i-10x)}$                  | $t = 0,649$  | $\int_{-5}^{-1} \sin(\lg(x^2))$        | 2,877   |
| $x = 6$      | $d = \sum_{i=-7}^0 x^2 + \prod_{j=1}^5 \left( \frac{j^2}{x^2} \sqrt{xj} \right)$                     | $d = 288,2$  | $\int_{14}^{21} \lg  e^x $             | 53,201  |
| $u = 45$     | $d = u + \sum_{j=10}^{18} \ln(uj) + \prod_{i=1}^9 \frac{1}{\sqrt{i+8}}$                              | $d = 102,9$  | $\int_{-17}^{-14} \sin^3(\sqrt[5]{q})$ | -2,887  |
| $x = \pi/20$ | $s = \sqrt{x^3} \cdot \sum_{u=0}^4 \prod_{j=15}^{20} \frac{\cos j + \sin u}{2x}$                     | $s = -9,352$ | $\int_2^5 x \sqrt{e^x - 1}$            | 72,187  |
| -            | $m = \left  \prod_{p=1}^4 \left( \frac{p}{16} \sum_{i=0}^8 \ln(i+6) \right) \right $                 | $m = 63,47$  | $\int_3^5 \lg^3 x + x^2$               | 33,108  |
| -            | $f = \sum_{i=1}^{20} \frac{\sin(2i+2)}{\cos(2i)} - \prod_{j=-7}^{-3} e^j$                            | $f = 18,233$ | $\int_3^{11} \sqrt[5]{\sin(3x)}$       | -0,359  |
| $j = 4$      | $k = \left  \sum_{w=-3}^1 jw \prod_{i=-6}^{-1} \left( 1 + \frac{w}{i} \right)^{\frac{1}{3}} \right $ | $k = 84,5$   | $\int_{21}^{27} \sin \cos \sqrt{x}$    | 1,056   |
| $y = 0,28$   | $h = \sum_{j=-5}^1 \left( j + \sin^2 j \prod_{b=1}^4 \sqrt{b} \right) y$                             | $h = 1,231$  | $\int_{-4}^0 e^{\sqrt{ \cos(x) }}$     | 9,105   |
| $x = 0,375$  | $a = \prod_{j=-5}^0 e^{xj} + \sum_{i=1}^7 (i \ln^3 x)$   | $a = -26,42$ | $\int_5^6 \sqrt{q} \cdot e^q$          | 602,312 |
| $t = \pi/3$  | $v = \prod_{j=-9}^{-1} \sin t \sum_{i=0}^{10} \frac{t \sin(2j)}{2i-1}$                               | $v = -0,015$ | $\int_0^1 \ln^3 \sin t$                | -6,042  |
| $x = 10$     | $r = \prod_{b=1}^{15} \frac{\ln(xb)}{10} + \sum_{m=15}^{25} \frac{1}{\ln(xm)}$                       | $r = 2,083$  | $\int_6^{13} \cos^2 x^3$               | 3,503   |

## Практическая работа №13. Работа с комплексными числами в Mathcad (часть 1)

**ЦЕЛЬ:** изучить ввод комплексных чисел в программе Mathcad и их расчет

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

Числа в Mathcad хранятся как числа двойной точности с плавающей точкой, которые при вводе и выводе могут отображаться:

в формате **целых чисел**, например, 345;

в виде **десятичной дроби** (в качестве разделителя используется точка), например, 2.456 ;

в виде **простой дроби**, например,  $\frac{1}{3}$  ;

в виде **смешанной дроби**, например,  $4\frac{3}{5}$  ;

в **экспоненциальной форме**, например,  $2.654 \cdot 10^{32}$  .

Можно указать формат отображения результатов расчетов. В Mathcad это обеспечивается посредством меню **Формат, Результат**. При этом появляется диалоговое окно, в котором указывается формат отображения результатов рис.26.

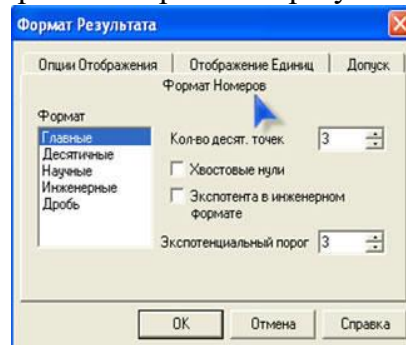


Рисунок 26

Пример использования различных форматов для отображения результатов расчетов представлен на листинге.

$$\frac{8}{7} = 1.14286 \quad \frac{8}{7} = 1.1429 \quad \frac{8}{7} = 1.143E+000 \quad \frac{8}{7} = 1.143 \times 10^0 \quad \frac{8}{7} = 1\frac{1}{7}$$

При работе с **комплексными числами** в Mathcad следует учитывать некоторые особенности организации вычислений:

- для **многозначной функции** программа возвращает значение, составляющее на комплексной плоскости минимальный положительный угол с положительным направлением действительной оси, например,  $\sqrt[3]{-16} = 1.414 + 1.414i$  ;

- следует помнить **специфику выполнения арифметических операций** в Mathcad, идентичных с точки зрения классической математики, но различающихся в написании. Подкоренное выражение рассматривается как действительное, а степень числа - как комплексное, например:

$$\sqrt[3]{-1} = -1 \quad \blacksquare, \text{ но } (-1)^{\frac{1}{3}} = 0.5 + 0.866i$$

В Mathcad существует **набор встроенных функций** для работы с комплексными числами, которые приведены на рис. 27.

| Обозначение в Mathcad | Описание   | Пример<br>$Z := 2.34 + 3.14i$ |
|-----------------------|--|-------------------------------|
| $Im(z)$               | Вычисление мнимой части комплексного числа   | $Im(z) = 3.14$                |
| $Re(z)$               | Вычисление действительной части комплексного числа   | $Re(z) = 2.34$                |
| $arg(z)$              | Определение значение угла, между радиусом – вектором и осью X  | $arg(z) = 0.93$               |
| $csgn(z)$             | Принимает значение=0, если $Z=0$ ; значение =1, если $Re(z) \geq 0$ и $Im(z) > 0$ ; значение -1 в других случаях | $csgn(z) = 1$                 |
| $signum(z)$           | Принимает значение 0, если $Z=0$ ; $Z/ Z $ - в других случаях.   | $signum(z) = 0.598 + 0.802i$  |

Рисунок 27

Кроме этого, при работе с комплексными числами следует учитывать, что  $|Z|$  определяет модуль комплексного числа, а  $\bar{Z}$  - число, комплексно сопряженное с  $Z$ . Например:

$$Z := 2.34 + 3.14i \quad |Z| = 3.916 \quad \bar{Z} = 2.34 - 3.14i$$

Чтобы ввести  $\bar{Z}$ , необходимо задать  $Z$ , а затем нажать клавишу  $[*]$ .

## ХОД РАБОТЫ

**Задание 1.** Найти значение выражения:

$$\frac{x-145}{y+5} \text{ при } x = 200, y = 5,$$

Для решения таких примеров используется знак присвоить «:=». Решим этот пример в MathCAD.

Важно, что MathCAD выполняет все последовательно, поэтому значения, которые принимает  $x, y$  необходимо вводить выше, чем сам пример.

Для написания дроби, используем панель «калькулятор» рис. 28

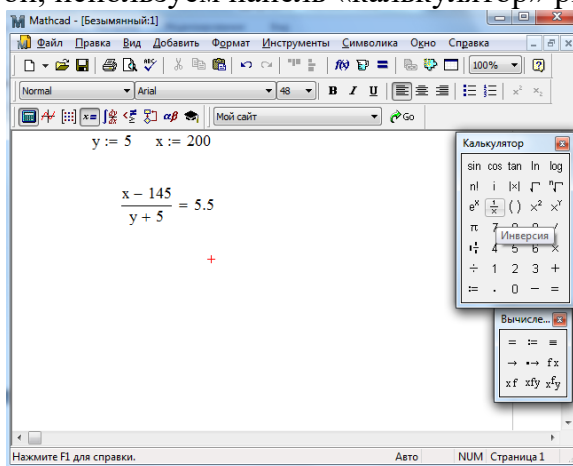


Рисунок 28

**Задание 2.** Найти значение выражения:

$$\sqrt{\frac{a^3}{b} - 2a + \frac{b}{a} - \frac{c}{ab}}, \text{ при } a = 25, b = 10$$

$$\left(\frac{a+b}{2a} - \frac{b}{a+b}\right) \left(\frac{a}{b} + \frac{b}{a} + 2\right), \text{ при } a = 0,75; b = \frac{4}{3}$$

$$\frac{4 - \sqrt[3]{a^2}}{(2 + \sqrt[3]{ab})^2 - (\sqrt[3]{a} + 2\sqrt[3]{b})^2}, \text{ при } a = \sqrt[3]{3}; b = \sqrt[3]{0,008}$$

## Практическая работа №13. Работа с комплексными числами в Mathcad (часть 2)

**ЦЕЛЬ:** изучить математические операции с комплексными числами

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

#### Пример 1. Математические операции

В MathCad поддерживаются все возможные операции над комплексными числами (сложение, вычитание, умножение, деление и др.).

Мнимая единица "i" вводится с панели инструментов "Calculator".

$$z1 := 6.85 + 9.4i$$

$$z2 := 2 - 0.5i$$

$$z1 + z2 = 8.85 + 8.9i$$

$$z1 - z2 = 4.85 + 9.9i$$

$$z1 \cdot z2 = 18.4 + 15.375i$$

$$\frac{z1}{z2} = 2.118 + 5.229i$$

Комплексное число может быть также аргументом функции, элементом матрицы, вектора:

$$\sin(z2) = 1.025 + 0.217i$$

$$A := \begin{pmatrix} 2 + 3i & 1 - 2i & 1.5 - 0.2i \\ 5i & 2 & 3 + i \end{pmatrix}$$

$$B := \begin{pmatrix} 1 - 2i & 5 \\ 4 & 2 + 2i \\ 4.2 & 3i \end{pmatrix}$$

$$A \cdot B = \begin{pmatrix} 18.3 - 9.84i & 16.6 + 17.5i \\ 30.6 + 9.2i & 1 + 38i \end{pmatrix}$$

#### Пример 2. Функция для обработки комплексных чисел.

$$z1 = 6.85 + 9.4i$$

$$\arg(z1) = 0.941$$

- Аргумент комплексного числа (радиан).

$$\operatorname{atan}\left(\frac{\operatorname{Im}(z1)}{\operatorname{Re}(z1)}\right) = 0.941$$

$$\psi := \arg(z1) \cdot \frac{180}{\pi}$$

$$\psi = 53.918$$

(градусов)

$$|z1| = 11.631$$

- Модуль комплексного числа.

$$\sqrt{\operatorname{Re}(z1)^2 + \operatorname{Im}(z1)^2} = 11.631$$

$$\operatorname{Im}(z1) = 9.4$$

- Мнимая часть комплексного числа.

$$\operatorname{Im}(z2) = -0.5$$

$$\operatorname{Re}(z1) = 6.85$$

- Действительная часть комплексного числа.

$$\operatorname{Re}(z2) = 2$$

$$z1 = 6.85 + 9.4i$$

$$|z1| \cdot e^{\arg(z1)i} = 6.85 + 9.4i$$

- Преобразование из алгебраической формы в экспоненциальную.

**Задание.** Решить примеры самостоятельно

| Задание  |
|--|
| $\frac{\left(68\frac{7}{30} - 66\frac{5}{18}\right) : 6\frac{1}{9} + \left(\frac{7}{40} + \frac{3}{32}\right) \cdot 4,5}{0,04}$  |
| $\frac{\left[\left(40\frac{7}{30} - 38\frac{5}{12}\right) : 10,9 + \left(\frac{7}{8} - \frac{7}{30}\right) \cdot 1\frac{9}{11}\right] \cdot 4,2}{0,008}$                                 |
| $\left[ \frac{\left(2,4 + 1\frac{5}{7}\right) \cdot 4,375}{\frac{2}{3} - \frac{1}{6}} - \frac{\left(2,75 - 1\frac{5}{6}\right) \cdot 21}{8\frac{3}{20} - 0,45} \right] : \frac{67}{200}$ |

### Практическая работа № 15. Расчет цепей постоянного тока

**ЦЕЛЬ:** Научиться рассчитывать цепи постоянного тока

#### ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

При расчете электрических цепей постоянного тока с одним источником питания часто необходимо проводить вычисления, связанные с нахождением эквивалентного сопротивления цепи, с определением токов во всех ветвях цепи и напряжений на источнике и на каждом приемнике отдельно.

Из условия задачи известны величина ЭДС аккумуляторной батареи  $E$ , ее внутреннее сопротивление  $R_0$  и сопротивления приемников  $R_1, R_2, R_3$ . На первом этапе расчета определяем эквивалентное сопротивление цепи  $R_c$ , воспользовавшись правилами последовательного и параллельного соединения пассивных элементов цепи. Затем, используя законы Ома для участка цепи, а также закон Кирхгофа для замкнутого контура, рассчитываем токи во всех ветвях цепи  $I, I_1, I_2$  и напряжения на отдельных участках  $U, U_{12}, U_3$  рис.29

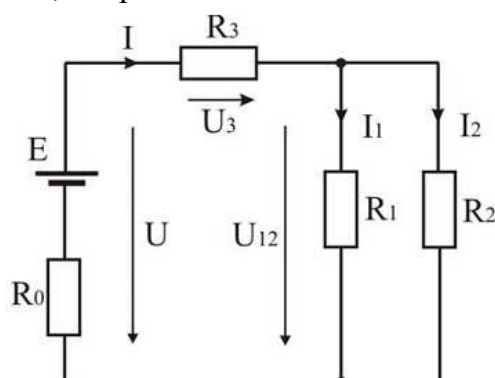


Рисунок 29



## ХОД РАБОТЫ

Значения исходных величин могут быть изменены, представленные формулы могут быть скорректированы в зависимости от характера электрической цепи, однако возможности программы MathCad позволяют автоматически производить необходимые вычисления, обновляя изменившиеся значения рис.30.

Исходные данные:  $R1 := 10$      $E := 78$   
 $R2 := 5$   
 $R3 := 4$   
 $R0 := 0.5$

1. Расчет эквивалентного сопротивления цепи  $R_c$ :

$$R_c := R0 + R3 + \frac{R1 \cdot R2}{R1 + R2}$$

$$R_c = 8$$

2. Ток в цепи источника  $I$  рассчитываем по закону Ома:

$$I := \frac{E}{R_c}$$

$$I = 10$$

3. Определение напряжений на отдельных участках цепи:

$$U_{12} := \left( \frac{R1 \cdot R2}{R1 + R2} \right) \cdot I \quad U3 := R3 \cdot I \quad U := E - I \cdot R0$$

$$U_{12} = 33$$

$$U3 = 40 \quad U = 73$$

4. Определение токов в ветвях цепи:

$$I1 := \frac{U_{12}}{R1} \quad I2 := \frac{U_{12}}{R2}$$

$$I1 = 3$$

$$I2 = 7$$

Рисунок 30

Проводя расчёт цепи в Mathcad нужно учитывать что использовать переменную которой присвоено значение можно только если она расположена справа от присвоения ей значения на том же уровне или снизу. Для примера можно провести расчёт в Matcad такой цепи рис.31:

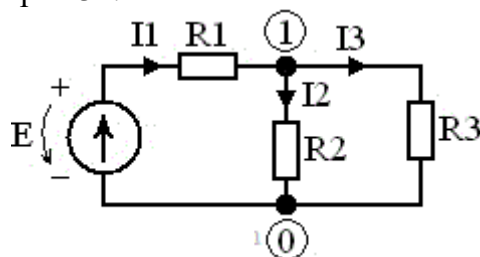


Рисунок 31

Для начала необходимо присвоить переменным значения параметров элементов цепи.

$R1 := 4$      $R2 := 2$      $R3 := 8$      $E := 28$

Из схемы рис.31 видно что резисторы  $R2$  и  $R3$  соединены параллельно их можно заменить эквивалентным сопротивлением  $R2 \parallel R3$ .

Чтобы вычислить это сопротивление можно сопротивления  $R2$  и  $R3$  заменить проводимостями  $G2$  и  $G3$ . Проводимость находится как отношение единицы к сопротивлению (т.е. проводимость - величина обратная сопротивлению). Сложив

проводимости получим эквивалентную проводимость  $G_{1||G2}$  из которой легко найти сопротивление  $R_{1||R2}$ .

$$G_{2||G3} := G_2 + G_3 \quad G_{2||G3} = 0.625 \text{ - проводимость цепи состоящей из параллельно соединённых резисторов } R_2 \text{ и } R_3$$

$$R_{2||R3} := \frac{1}{G_{2||G3}} \quad R_{2||R3} = 1.6 \text{ - сопротивление цепи состоящей из параллельно соединённых резисторов } R_2 \text{ и } R_3$$

Получившуюся схему можно преобразовать к виду рис.32:

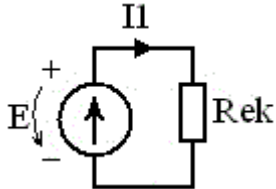


Рисунок 32

Где  $R_{ek}$  - эквивалентное сопротивление всей цепи которое находится суммированием сопротивлений  $R_{2||R3}$  и  $R_1$ . Из этой схемы можно найти ток  $I_1$  который, по первому закону Кирхгофа, равен сумме  $I_2+I_3$ . Найдя этот ток, по закону Ома, можно определить потенциал узла 1, из второй схемы, умножив  $I_1$  на  $R_{2||R3}$ . Зная потенциал узла 1, из первой схемы, можно найти токи второго и третьего резисторов учитывая что потенциал узла 0 равен нулю и напряжение на этих резисторах равно разности потенциалов 1 и 0.

$$R_{ek} := R_1 + R_{2||R3} \quad R_{ek} = 5.6 \text{ - эквивалентное сопротивление всей цепи}$$

$$I_1 := \frac{E}{R_{ek}} \quad I_1 = 5 \text{ - ток резистора } R_1$$

$$\phi_1 := I_1 \cdot R_{2||R3} \quad \phi_1 = 8 \text{ - потенциал первого узла}$$

$$I_2 := \phi_1 \cdot G_2 \quad I_2 = 4 \text{ - ток резистора } R_2$$

$$I_3 := \phi_1 \cdot G_3 \quad I_3 = 1 \text{ - ток резистора } R_3$$

После нахождения всех токов необходимо проверить баланс мощностей.

$$I_1^2 \cdot R_1 + I_2^2 \cdot R_2 + I_3^2 \cdot R_3 = 140$$

$$I_1 \cdot E = 140$$

Если сумма мощностей резисторов равна сумме мощностей источников то баланс мощностей выполнен и расчёт произведён правильно.

## Практическая работа 16. Сравнение результатов работы в Mathcad с результатами моделирования в Multisim

**Цель:** Сравнить результаты моделирования последовательного и параллельного соединения резисторов

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

Элемент электрической цепи, предназначенный для использования его электрического сопротивления, называется резистором. Применяют последовательные, параллельные и смешанные соединения резисторов

А) При последовательном соединении по всем элементам протекает один и тот же ток:

$$I_{об} = I_1 = I_2 = I_3$$

Б) Эквивалентное сопротивление цепи равно сумме сопротивлений, включенных резисторов:  $R_{об} = R_1 + R_2 + R_3$

В) Общее сопротивление цепи равно сумме напряжения всех резисторов:  $U_{об} = U_1 + U_2 + U_3$

ПА) При параллельно соединении резисторов на зажимах элементов имеется одно и тоже напряжение:  $U_{об} = U_1 = U_2 = U_3$

Б) Эквивалентное сопротивление рассчитывается по формуле:  $\frac{1}{R_{об}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$

В) Величина тока определяется по первому закону Кирхгофа  $I_{об} = I_1 + I_2 + I_3$ .

Узлом называется точка цепи, к которой подходит не менее трёх проводов. Ветвью называется часть цепи заключенная между двумя узлами  $I_1; I_2; I_3$  - токи ветвей.

$I$  – ток, проходящий в узел.

1. Собрать схему рис.33 с последовательным соединением резисторов в программе Multisim

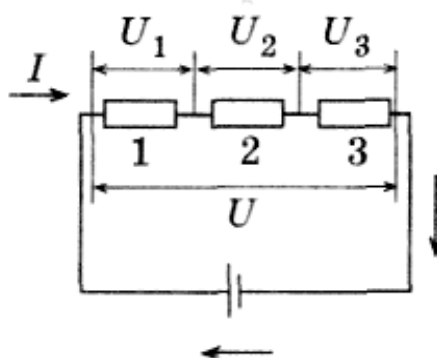


Рисунок 33

2. Запустить пробник, записать показания приборов в таблицу 20.

3. Собрать схему рис.34 с параллельным соединением резисторов в программе Multisim.

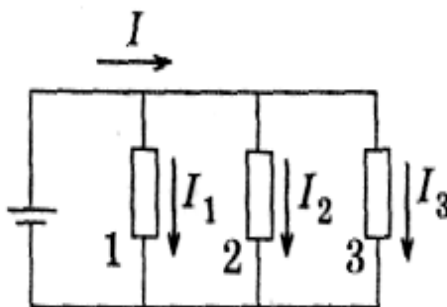


Рисунок 34

4. Запустить пробник, записать показания приборов в таблицу 21.

5. Рассчитать общие сопротивления при последовательном и параллельном соединениями резисторов в программе Mathcad.

Опыт № 1. \_\_\_\_\_  
(Укажите соединение резисторов)

$$R_1 = \underline{\hspace{2cm}}; \quad R_2 = \underline{\hspace{2cm}};$$

$$R_{об} = \underline{\hspace{2cm}};$$

$$U_{об} = \underline{\hspace{2cm}}.$$

Опыт № 2. \_\_\_\_\_  
(Укажите соединение резисторов)

$$R_1 = \underline{\hspace{2cm}}; \quad R_2 = \underline{\hspace{2cm}};$$

$$R_{об} = \underline{\hspace{2cm}};$$

$$U_{об} = \underline{\hspace{2cm}}.$$

Таблица 21

| № опыта | Измерить |       |     |     | Вычислить |       |          |          |
|---------|----------|-------|-----|-----|-----------|-------|----------|----------|
|         | $U_1$    | $U_2$ | $U$ | $I$ | $R_1$     | $R_2$ | $R_{об}$ | $U_{об}$ |
|         | $B$      | $B$   | $B$ | $A$ | $Ом$      | $Ом$  | $Ом$     | $B$      |
| 1       |          |       |     |     |           |       |          |          |
| 2       |          |       |     |     |           |       |          |          |

### Практическая работа 17. Расчет цепей переменного тока.

Цель: Научиться рассчитывать цепи постоянного тока MathCad

#### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

При проведении электротехнических расчетов разветвленных и неразветвленных цепей переменного тока удобно использовать все известные методы анализа цепей постоянного тока. В этом случае применяется комплексный метод расчета.

С помощью средств программы MathCad возможно производить расчеты, используя величины, выраженные в комплексном виде.

Особенностью формы записи комплексных чисел в программе MathCad является следующее: при вводе комплексных чисел нельзя использовать  $i$  или  $j$  сами по себе, необходимо всегда печатать  $i$  или  $j$ .

В MathCad существуют следующие специальные функции и операторы для работы с комплексными числами:

$\text{Re}(z)$  — вещественная часть;

$\text{Im}(z)$  — мнимая часть;

$\text{arg}(z)$  — угол в комплексной плоскости между вещественной осью и  $z$ .

Возвращает результат между  $-\pi$  и  $\pi$  радиан;

$|z|$  — модуль  $z$ . Чтобы записать модуль от выражения, заключите его в выделяющую рамку и нажмите клавишу с вертикальной полосой  $|$ .

$\bar{z}$  — число, комплексно сопряженное к  $z$ . Чтобы применить к выражению оператор сопряжения, выделите выражение, затем нажмите двойную кавычку ( $''$ ).

На рис. 35 представлен пример перевода из алгебраической формы записи комплексного числа в показательную форму и обратно, что необходимо при выполнении электротехнических расчетов цепей переменного тока.

Перевод комплексной величины, представленной в алгебраической форме, в показательную форму.

$$r := 145.082 \quad x := -35.551$$

$$z := r + i \cdot x$$

$$z = 145.082 - 35.551i$$

$$|z| = 149.374$$

$$\arg(z) = -0.24$$

$$\phi := \frac{\arg(z) \cdot 180}{\pi}$$

$$\phi = -13.769$$

Перевод комплексной величины, представленной в показательной форме, в алгебраическую форму.

$$\beta := \frac{-18 \cdot \pi}{180}$$

$$\alpha := \frac{19 \cdot \pi}{180}$$

$$u := 88.23 \cdot e^{i \cdot \beta}$$

$$u1 := 55.14 \cdot e^{i \cdot \alpha}$$

$$u = 83.912 - 27.265i$$

$$u1 = 52.136 + 17.952i$$

$$\operatorname{Re}(u) = 83.912$$

$$\operatorname{Re}(u1) = 52.136$$

$$\operatorname{Im}(u) = -27.265$$

$$\operatorname{Im}(u1) = 17.952$$

Рисунок 35

Приведем пример расчета сложной цепи переменного тока, представленной на рис. 36. Исходными данными для задачи являются величины сопротивлений и напряжения на входе схемы.

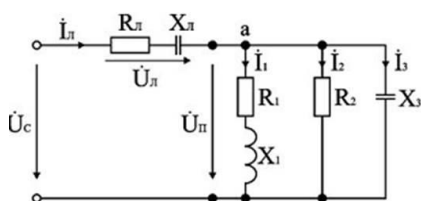


Рисунок 36

Расчет цепи произведем поэтапно. Сначала рассчитаем проводимости параллельного участка цепи, затем полное сопротивление цепи. Этот расчет представлен на рис. 37

Исходные данные:  $r1 := 4$        $r2 := 8$        $r3 := 0$        $г1 := 1.8$   
 $Uc := 660e^{i \cdot 0}$        $x1 := 3$        $x2 := 0$        $x3 := -16$        $х1 := -0.6$   
 $z1 := r1 + i \cdot x1$      $z2 := r2 + i \cdot x2$      $z3 := r3 + i \cdot x3$      $zл := г1 + i \cdot х1$

1. Определение активной и реактивной проводимостей:

$$q := \frac{r1}{(|z1|)^2} + \frac{r2}{(|z2|)^2} + \frac{r3}{(|z3|)^2} \quad b := \frac{x1}{(|z1|)^2} + \frac{x2}{(|z2|)^2} + \frac{x3}{(|z3|)^2}$$

$$q = 0.29 \qquad b = 0.06$$

2. Определение проводимости параллельного участка цепи:

$$Y := \sqrt{b^2 + q^2} \qquad Y = 0.29$$

3. Определение активного, реактивного и полного сопротивления схемы замещения параллельного участка:

$$zп := \frac{1}{Y} \qquad гп := q \cdot zп^2 \qquad хп := b \cdot zп^2$$

$$zп = 3.44 \qquad гп = 3.37 \qquad хп = 0.68$$

4. Определение полного сопротивления схемы замещения цепи:

$$z := (гл + гп) + i \cdot (хл + хп) \quad z = 5.17 + 0.08i$$

### Рисунок 37

Воспользовавшись полученными данными, определим все токи и напряжения в цепи, используя законы Ома и Кирхгофа для цепей переменного тока. Расчет токов и напряжений представлен на рис.38.

5. Определение тока во всей цепи:

$$Iл := \frac{Uc}{z} \qquad Iл = 127.59 - 1.98i$$

$$|Iл| = 127.61 \qquad \phiл := \frac{\arg(Iл) \cdot 180}{\pi} \qquad \phiл = -0.89$$

6. Определение напряжений на отдельных участках цепи:

$$Uл := zл \cdot Iл \qquad Uл = 228.48 - 80.12i$$

$$|Uл| = 242.12 \qquad \phiлл := \frac{\arg(Uл) \cdot 180}{\pi} \qquad \phiлл = -19.32$$

$$Uп := (гп + i \cdot хп) \cdot Iл \qquad Uп = 431.52 + 80.12i$$

$$|Uп| = 438.9 \qquad \phiп := \frac{\arg(Uп) \cdot 180}{\pi} \qquad \phiп = 10.52$$

7. Нахождение токов в параллельных ветвях цепи:

$$I1 := \frac{Uп}{z1} \qquad I2 := \frac{Uп}{z2} \qquad I3 := \frac{Uп}{z3}$$

$$I1 = 78.66 - 38.96i \qquad I2 = 53.94 + 10.01i \qquad I3 = -5.01 + 26.97i$$

$$|I1| = 87.78 \qquad |I2| = 54.86 \qquad |I3| = 27.43$$

$$\phi1 := \frac{\arg(I1) \cdot 180}{\pi} \qquad \phi2 := \frac{\arg(I2) \cdot 180}{\pi} \qquad \phi3 := \frac{\arg(I3) \cdot 180}{\pi}$$

$$\phi1 = -26.35 \qquad \phi2 = 10.52 \qquad \phi3 = 100.52$$

### Рисунок 38

На следующем этапе расчета цепи определим активные, реактивные и полные мощности источника и всех приемников, произведем расчет баланса мощностей. Данные расчета показаны на рис. 39.

8. Определение активной, реактивной и полной мощностей отдельных участков цепи и источника:

|   |   |
|---|---|
| $P_1 := U_n \cdot I_1 \cdot \cos(\arg(z_1))$          | $Q_1 := U_n \cdot I_1 \cdot \sin(\arg(z_1))$  |
| $P_1 = 2.97 \times 10^4 - 8.41i \times 10^3$          | $Q_1 = 2.22 \times 10^4 - 6.31i \times 10^3$  |
| $P_2 := U_n \cdot I_2 \cdot \cos(\arg(z_2))$          | $Q_2 := U_n \cdot I_2 \cdot \sin(\arg(z_2))$  |
| $P_2 = 2.25 \times 10^4 + 8.64i \times 10^3$          | $Q_2 = 0$                                     |
| $P_3 := U_n \cdot I_3 \cdot \cos(\arg(z_3))$          | $Q_3 := U_n \cdot I_3 \cdot \sin(\arg(z_3))$  |
| $P_3 = -2.65 \times 10^{-13} + 6.88i \times 10^{-13}$ | $Q_3 = 4.32 \times 10^3 - 1.12i \times 10^4$  |
| $P_l := U_l \cdot I_l \cdot \cos(\arg(z_l))$          | $Q_l := U_l \cdot I_l \cdot \sin(\arg(z_l))$  |
| $P_l = 2.75 \times 10^4 - 1.01i \times 10^4$          | $Q_l = -9.17 \times 10^3 + 3.38i \times 10^3$ |
| $S_1 := U_n \cdot I_1$                                | $P_c := U_c \cdot I_l \cdot \cos(\arg(z))$    |
| $S_1 = 3.71 \times 10^4 - 1.05i \times 10^4$          | $P_c = 8.42 \times 10^4 - 1.31i \times 10^3$  |
| $S_2 := U_n \cdot I_2$                                | $Q_c := U_c \cdot I_l \cdot \sin(\arg(z))$    |
| $S_2 = 2.25 \times 10^4 + 8.64i \times 10^3$          | $Q_c = 1.31 \times 10^3 - 20.26i$             |
| $S_3 := U_n \cdot I_3$                                | $S_c := U_c \cdot I_l$                        |
| $S_3 = -4.32 \times 10^3 + 1.12i \times 10^4$         | $S_c = 8.42 \times 10^4 - 1.31i \times 10^3$  |
| $S_l := U_l \cdot I_l$                                |   |
| $S_l = 2.9 \times 10^4 - 1.07i \times 10^4$           |   |

9. Баланс мощностей:

$$S := S_1 + S_2 + S_3 + S_l$$

$$S = 8.42 \times 10^4 - 1.31i \times 10^3$$

Рисунок 39

Сравниваем значения  $S_c$  (полная мощность источника синусоидального тока) и  $S$  (полная мощность всех приемников цепи) и убеждаемся в том, что баланс мощностей «сошелся». Это означает, что задача решена верно.

## Практическая работа 18. Сравнение результатов расчетов в Mathcad с результатами моделирования в NI Multisim

**Цель:** Сравнить результаты моделирования соединения катушки индуктивности и конденсатора

1. Собрать схему по рис.40 в программе NI Multisim

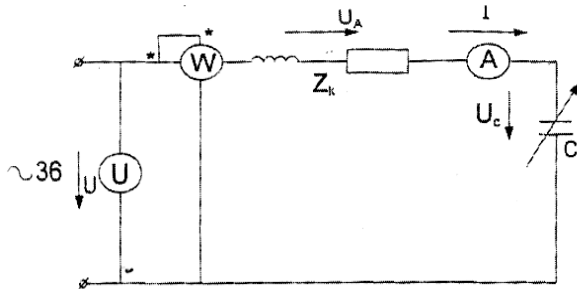


Рисунок 40

2.

3. Включить ёмкость  $C$  в пределах от 30 до 50 мкФ.
4. Запустить пробник и измерить параметры схемы и записать их в табл.22.
5. Подобрать ёмкость так, чтобы ток в цепи был наибольший.
6. Записать в таблицу показания приборов.
7. Включить  $C = 94.75$  мкФ и вновь повторить опыт.
8. Запустить программу Mathcad, вычислить по формулам:

$$S = U \cdot I; \quad S = \underline{\hspace{2cm}}; \quad Z = \frac{U}{I}; \quad Z = \underline{\hspace{2cm}};$$

$$P = I^2 \cdot r; \quad r = \underline{\hspace{2cm}}; \quad \cos \varphi = \frac{P}{S}; \quad \cos \varphi = \underline{\hspace{2cm}};$$

$$U_a = \frac{P}{I}; \quad U_a = \underline{\hspace{2cm}}; \quad U_L^2 = U_K^2 - U_A^2; \quad U_L = \underline{\hspace{2cm}}.$$

9. Сравнить результаты моделирования в программе Multisim и Mathcad

Таблица 22

| № опыта | Измерить |     |     |     |       |       | Вычислить |     |     |                |       |       |
|---------|----------|-----|-----|-----|-------|-------|-----------|-----|-----|----------------|-------|-------|
|         | $C$      | $U$ | $I$ | $P$ | $U_C$ | $U_K$ | $S$       | $Z$ | $r$ | $\cos \varphi$ | $U_a$ | $U_L$ |
|         | мкФ      | В   | А   | Вт  | В     | В     | ВА        | Ом  | Ом  | -              | В     | В     |
| 1       |          |     |     |     |       |       |           |     |     |                |       |       |
|         |          |     |     |     |       |       |           |     |     |                |       |       |
| 2       |          |     |     |     |       |       |           |     |     |                |       |       |

## Практическая работа 19. Язык программирования C/C++. Идентификаторы.

**Цель:** Изучить язык программирования C++.

**Задание 1.** При выполнении программы:

**Пример 1.**

```
#include <stdio.h>
```



```

void main()
{ float a,b,c;
  int m,n;
  scanf("%f%d%f%d%f",&a,&m, &b, &n, &c);
  printf("\na=%8.3fd=%7.2fc=%12.3e",a,b,c);
  printf("\nm=%10dn=%5d",m,n);
}

```

с клавиатуры была введена следующая символьная последовательность:  
32.4 87 0.05 4567 2314.45 <Enter>

Как будет выглядеть на экране результат работы программы?

Для приводимых примеров будем приводить только текст программы и результат ее выполнения. Программы должны быть написаны в Visual Studio.

**Пример 2.** Напишите программу вычисления алгебраических выражений с приведением типов char, int, float, double.

Возможный программный код решения примера:

```

#include <stdio.h>
#include <conio.h>
#include <math.h>

int main (void) {
char ch;
int a, b;
float x;
double y, z;
ch = 'C';
a = 2; b = 5;
x = 5.5F;
y = 6.0 ;
z = ch + a + b + x + y;

// Результат приведения типов
printf("\n\t The result of the conversion types: %lf\n", z);

// Явное преобразование типов
z = (double) ch + (double) a + (double) b + (double) x + y;

printf("\n\t The apparent conversion types: %lf\n", z);
z = sqrt((double)a/b);
printf("\n\t z = sqrt(%d/%d) = %lf\n", a, b, z);
printf("\n\t z = lg(%d/%d) = %lf\n", a, b, log10(z));
printf("\n\t z = ln(%d/%d) = %lf\n", a, b, log(z));

printf("\n Press any key: ");
_getch();
return 0;
}

```

В программу включена библиотека математических функций `math.h`, в которой `sqrt()` – функция извлечения квадратного корня, `log10()` – логарифмическая функция по основанию 10, `log()` – функция натурального логарифма. Все эти функции возвращают результат типа `double` и вычисляют от числа (выражения) также типа `double`.

## Задание 2

1. В качестве вводимого символа используйте первую букву своей фамилии (буквами латинского алфавита). В отчет вставьте полученный результат.
2. Переменную `u` задайте в виде `u = 6`.
3. Вычислите заданный квадратный корень без явного приведения типов.
4. Определите переменные с суффиксами `U`, `F`, `L` и произведите с ними арифметические действия.
5. Вычислите логарифм от числа  $10^X$  по основанию  $2^X$ , где `X` – номер компьютера, за которым выполняется лабораторная работа. Выполните тестовый пример для проверки вычислений, т.е. для заданного основания подберите число, от которого логарифм с заданным основанием даст целое число. Например,  $\log_{10}(100.0) = 2.0000$ .

**Пример 3.** Напишите программу вычисления площади круга и его длины окружности по заданному радиусу, вводимого пользователем с клавиатуры, а также вывода на консоль максимальных значений чисел типа `int`, `float` и `double`.

Для решения примера следует воспользоваться математической библиотекой компилятора, т.е. включить в программу заголовочный файл `<math.h>`, а также заголовочные файлы `<limits.h>`, `<float.h>`.

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>

// Для числа пи
#define _USE_MATH_DEFINES
#include <math.h>

#include <limits.h>
#include <float.h>

int main (void)
{
    double R;
    double Sr, Lr;

    printf("\n Enter a real greater than zero: ");
    scanf_s("%lf", &R);
    Sr = M_PI*R*R;
    Lr = 2*M_PI*R;
    printf("\n Area of a circle of radius R = %g is %g", R, Sr);
    printf("\n The length of a circle of radius R = %g is %g",R,Lr);
    puts("");

    printf("\n Maximum integer: %d\n ", INT_MAX);
```

```
printf(" Maximum real number of float: %g\n ", FLT_MAX);
printf("Maximum real number type double: %g\n ", DBL_MAX);
```

```
printf("\n Press any key: ");
_getch();
return 0;
}
```

В программу включена константа `_USE_MATH_DEFINES` для работы с числом `M_PI` ( $\pi$ ). Остальные константы можно найти в справочной документации компилятора. Например, через меню Help → Index системы MS Visual Studio 2008.

Функция `scanf_s()` определена в компиляторе языка C системы MS Visual Studio 2008. С этой функцией компилятор не выдает предупреждений.

### Задание 3

1. Напишите программу по вычислению площади эллипса. Данные для расчета должны вводиться с клавиатуры пользователем.

2. Рассчитайте величины  $\lg(\exp(100))$  и  $\ln(\exp(100))$ , пользуясь стандартными функциями математической библиотеки (`math.h`).

## Практическая работа 20. Язык программирования C/C++ Операторы. Массивы.

**Цель:** изучить операторы и массивы программ на C++.

### ХОД РАБОТЫ

**Задание 1.** При выполнении программы:

#### Пример 1.

```
#include <stdio.h>
void main()
{ float a,b,c;
  int m,n;
  scanf("%f%d%f%d%f",&a,&m, &b, &n, &c);
  printf("\na=%8.3fd=%7.2fc=%12.3e",a,b,c);
  printf("\nm=%10dn=%5d",m,n);
}
```

с клавиатуры была введена следующая символьная последовательность:  
32.4 87 0.05 4567 2314.45 <Enter>

Как будет выглядеть на экране результат работы программы?

Для приводимых примеров будем приводить только текст программы и результат ее выполнения. Программы должны быть написаны в Visual Studio.

**Пример 2.** Напишите программу вычисления алгебраических выражений с приведением типов `char`, `int`, `float`, `double`.

Возможный программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
```

```
int main (void) {
char ch;
int a, b;
float x;
double y, z;
```

```

ch = 'C';
a = 2; b = 5;
x = 5.5F;
y = 6.0 ;
z = ch + a + b + x + y;

// Результат приведения типов
printf("\n\t The result of the conversion types: %lf\n", z);

// Явное преобразование типов
z = (double) ch + (double) a + (double) b + (double) x + y;

printf("\n\t The apparent conversion types: %lf\n", z);
z = sqrt((double)a/b);
printf("\n\t z = sqrt(%d/%d) = %lf\n", a, b, z);
printf("\n\t z = lg(%d/%d) = %lf\n", a, b, log10(z));
printf("\n\t z = ln(%d/%d) = %lf\n", a, b, log(z));

    printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В программу включена библиотека математических функций `math.h`, в которой `sqrt()` – функция извлечения квадратного корня, `log10()` – логарифмическая функция по основанию 10, `log()` – функция натурального логарифма. Все эти функции возвращают результат типа `double` и вычисляют от числа (выражения) также типа `double`.

## Задание 2

6. В качестве вводимого символа используйте первую букву своей фамилии (буквами латинского алфавита). В отчет вставьте полученный результат.
7. Переменную `y` задайте в виде  $y = 6$ .
8. Вычислите заданный квадратный корень без явного приведения типов.
9. Определите переменные с суффиксами `U`, `F`, `L` и произведите с ними арифметические действия.
10. Вычислите логарифм от числа  $10^X$  по основанию  $2^X$ , где  $X$  – номер компьютера, за которым выполняется лабораторная работа. Выполните тестовый пример для проверки вычислений, т.е. для заданного основания подберите число, от которого логарифм с заданным основанием даст целое число. Например,  $\log_{10}(100.0) = 2.0000$ .

**Пример 3.** Напишите программу вычисления площади круга и его длины окружности по заданному радиусу, вводимого пользователем с клавиатуры, а также вывода на консоль максимальных значений чисел типа `int`, `float` и `double`.

Для решения примера следует воспользоваться математической библиотекой компилятора, т.е. включить в программу заголовочный файл `<math.h>`, а также заголовочные файлы `<limits.h>`, `<float.h>`.

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>

// Для числа пи
#define _USE_MATH_DEFINES
#include <math.h>

#include <limits.h>

```

```

#include <float.h>

int main (void)
{
double R;
double Sr, Lr;

printf("\n Enter a real greater than zero: ");
scanf_s("%lf", &R);
Sr = M_PI*R*R;
Lr = 2*M_PI*R;
printf("\n Area of a circle of radius R = %g is %g", R, Sr);
printf("\n The length of a circle of radius R = %g is %g",R,Lr);
puts("");

printf("\n Maximum integer: %d\n ", INT_MAX);
printf(" Maximum real number of float: %g\n ", FLT_MAX);
printf("Maximum real number type double: %g\n ", DBL_MAX);

printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В программу включена константа `_USE_MATH_DEFINES` для работы с числом `M_PI` ( $\pi$ ). Остальные константы можно найти в справочной документации компилятора. Например, через меню Help → Index системы MS Visual Studio 2008.

Функция `scanf_s()` определена в компиляторе языка C системы MS Visual Studio 2008. С этой функцией компилятор не выдает предупреждений.

### Задание 3

3. Напишите программу по вычислению площади эллипса. Данные для расчета должны вводиться с клавиатуры пользователем.
4. Рассчитайте величины  $\lg(\exp(100))$  и  $\ln(\exp(100))$ , пользуясь стандартными функциями математической библиотеки (`math.h`).

## Практическая работа 21. Ввод и вывод данных. Первая программ

**Цель:** изучить работу команды ввода и вывода данных. Написать первую программу

### ХОД РАБОТЫ

В языке СИ любая программа состоит из одной или более функций, задающих действия, которые нужно выполнить. Выполнение любой программы начинается с функции `main`. Далее идет текст программы, заключенный в фигурные скобки. Таким образом, структура программы имеет вид:

```

main ( )
{
Тело программы
}

```

В самом простом случае функция `main` не имеет аргументов, поэтому в скобках ничего не содержится. Для работы программы, обеспечивающей ввод и вывод информации, перед функцией `main` необходимо поместить строку:

```

#include <stdio.h>

```

## Алфавит языка и типы данных

Алфавит языка включает латинские прописные и строчные буквы, цифры и специальные знаки. К последним относятся: . (точка), , (запятая), ' (апостроф), : (двоеточие) и др.

Важным понятием языка является идентификатор, который используется в качестве имени объекта, например, переменной, функции и т.п. Идентификатор может содержать до 32 символов и состоит из букв и цифр, но начинается обязательно с буквы. Строчные буквы отличаются от прописных, поэтому идентификаторы SIGMA и sigma считаются разными.

В языке СИ существует несколько типов данных. Каждый тип данных определяется одним из следующих ключевых слов:

1. int (целый) – задает значения, к которым относятся все целые числа. Диапазон возможных целых значений лежит в пределах от  $-32768$  до  $32767$ , переменная типа int занимает 16 бит;

2. short (короткий целый) – соответствующие объекты не могут быть больше, чем int, переменные этого типа занимают 16 бит;

3. long (длинный целый) – соответствующие объекты не могут быть меньше, чем int. Переменная типа long занимает 32 бита и позволяет представить целые числа от  $-2147483648$  до  $2147483647$ ;

4. char (символьный) – задает значения, которые представляют различные символы;

5. unsigned (беззнаковый) – в языке СИ можно объявлять некоторые типы (char, short, int, long) беззнаковыми с помощью модификатора unsigned (например, unsigned short). Это значит, что соответствующие переменные не будут иметь отрицательных значений. В результате они могут принимать большие положительные значения, чем переменные знаковых типов. В случае типа int объявления вида «unsigned int a;» можно записать «unsigned a;»;

6. float (вещественный) – задает значения, к которым относятся вещественные числа, имеющие дробную часть, отделяемую точкой. Вещественные числа могут быть записаны также в экспоненциальной форме. Например,  $-1.58e+2$  (что равно  $-1,58 \cdot 10^2$ ). В языке СИ переменная типа float занимает 32 бита. Она может принимать значения в диапазоне от  $+3.4e-38$  до  $+3.4e+38$ ;

7. double (двойная точность) – определяет вещественные переменные, занимающие в два раза больше места, чем переменная типа float. Переменная типа double занимает 64 бита. Она может принимать значения в диапазоне от  $+1.7e-308$  до  $+1.7e+308$ .

## Ввод и вывод информации

### Форматный вывод

Вначале рассмотрим функцию, определяющую форматный вывод:

```
printf("управляющая строка", аргумент1, аргумент2, ... );
```

Управляющая строка содержит объекты трех типов: обычные символы, которые просто выводятся на экран дисплея, спецификации преобразования, каждая из которых вызывает вывод на экран значения очередного аргумента из последующего списка и управляющие символы-константы.

Каждая спецификация преобразования начинается со знака % и заканчивается некоторым символом, задающим преобразования.

Символ преобразования связан с типом переменных. приведем символы преобразования:

- 1) d – значением аргумента является десятичное целое число;
- 2) o – значением аргумента является восьмеричное целое число;
- 3) x – значением аргумента является шестнадцатеричное целое число;

4) с – значением аргумента является символ;  
5) s – значением аргумента является строка символов;  
6) e – значением аргумента является вещественное число в экспоненциальной форме;

7) f – значением аргумента является вещественное десятичное число с плавающей точкой;

8) u – значением аргумента является беззнаковое целое число;

9) p – значением аргумента является указатель (адрес).

Если после знака % записан не символ, то он выводится на экран. Функция printf использует управляющую строку, чтобы определить, сколько всего аргументов и каковы их типы.

Например, в результате работы программы получены переменная i, имеющая значение 100, и переменная j, имеющая значение 25. Обе переменные целого типа. Для вывода этих переменных на экран в виде

```
i=100 j=25
```

необходимо применить функцию

```
printf("i=%d j=%d",i,j);
```

Как было описано выше, в кавычках задается формат вывода. перед знаком % записываются символы, которые будут непосредственно выданы на экран. После знака % применена спецификация d, т.к. переменные i и j имеют целый тип. Сами i и j приведены через запятую в списке аргументов. Если результат должен быть представлен в виде

```
i=100; j=25
```

необходимо применить функцию

```
printf("i=%d; j=%d, i, j);
```

Если после знака % стоит цифра, то она задает поле, в котором будет выполнен вывод числа. Приведем несколько функций printf, которые будут обеспечивать вывод одной и той же переменной S целого типа, имеющей значение 336.

Функция printf(“%2d”, S); выдает на экран:

```
336
```

В этом примере ширина поля ( она равна двум) меньше, чем число цифр в числе 336, поэтому поле автоматически расширяется до необходимого размера.

Функция printf(“%6d”, S);

выдаст на экран:

```
__ _336
```

(6 позиций)

То есть, в результате работы функции число сдвинуто к правому краю поля, а лишние позиции перед числом заполнены пробелами.

Функция printf(“%-6d”, S);

выдаст на экран:

```
336__ _
```

(6 позиций)

Знак «минус» перед спецификацией приводит к сдвигу числа к левому краю поля.

Рассмотрим вывод вещественных чисел.

Если перед спецификацией f ничего не указано, то выводится число с шестью знаками после запятой. при печати числа с плавающей точкой перед спецификацией f тоже могут находиться цифры.

Рассмотрим на конкретном примере три возможные ситуации:

%bf – печать числа с плавающей точкой в поле из шести позиций;

`%.2f` – печать числа с плавающей точкой с двумя цифрами после десятичной точки;

`%6.2f` – печать числа с плавающей точкой в поле из шести позиций и двумя цифрами после десятичной точки.

Например, в результате работы программы получены переменные вещественного типа `a=3,687` и `b=10,17`.

Если для вывода значений использована функция

```
printf(“%7f %8f”,a,b);
```

то результат будет представлен в виде строки:

```
__ 3.687 ___ 10.17
```

(7 поз.) (8 позиций)

Как видно из примера, лишние позиции заполняются пробелами. Если для вывода значений использована функция

```
printf(“%.2f %/2f”, a, b);
```

то результатом будет строка:

```
3.69 10.17 ,
```

из которой следует, что в первом числе третья цифра после десятичной точки отброшена с округлением, т.к. указан формат числа с двумя цифрами после десятичной точки.

Если для вывода значений использована функция

```
printf(“%7.2f e”,a,b);
```

то будет выведена строка:

```
___ 3.681.010000e+01
```

(7 позиций)

Поскольку для вывода значения переменной `b` применена спецификация `e`, результат выдан в экспоненциальной форме. Следует отметить, что, если ширина поля меньше, чем число цифр в числе, то поле автоматически расширяется до необходимого размера.

Как было отмечено выше, в управляющей строке могут содержаться управляющие символьные константы. Среди управляющих символьных констант наиболее часто используются следующие:

- 1) `\a` – для кратковременной подачи звукового сигнала;
- 2) `\b` – для перевода курсора влево на одну позицию;
- 3) `\n` – для перехода на новую строку;
- 4) `\r` – для перевода курсора в начало текущей строки;
- 5) `\t` – для горизонтальной табуляции;
- 6) `\v` – для вертикальной табуляции.

Предположим, в результате работы программы переменная `i` получила значение 50. В результате записи инструкции вызова функции

```
printf(“\t ЭВМ\n%d\n”,i);
```

сначала выполнится горизонтальная табуляция (`\t`), т.е. курсор сместится от края экрана на 8 позиций, затем на экран будет выведено слово “ЭВМ”, после этого курсор переместится в начало следующей строки (`\n`), затем будет выведено целое значение `i` по формату `d`, и окончательно курсор перейдет в начало новой строки (`\n`). Таким образом, результат работы этой функции на экране будет иметь вид:

```
_____ ЭВМ
```

```
50
```

Для форматного ввода данных используется функция

```
scanf(«управляющая строка», аргумент1, аргумент2,...);
```

Если в качестве аргумента используется переменная, то перед ее именем записывается символ `&`.



Управляющая строка содержит спецификации преобразования и используется для установления количества и типов аргументов. спецификации для определения типов аргументов такие же, как и для функции printf. Перед символами d,o,x,f может стоять буква l. В первых трех случаях соответствующие переменные должны иметь тип long, а в последнем double.

Рассмотрим пример. Требуется ввести значения для переменных i (целого типа) и a (вещественного типа). Эту задачу выполнит функция:

```
scanf(“%d%f”,&i,&a);
```

В управляющей строке спецификации трех типов могут быть отделены друг от друга различными знаками, в том числе и пробелом. Следовательно, при занесении значений переменных необходимо использовать указанный разделитель. Если спецификации не отделены одна от другой никакими значениями, то значения переменных заносятся через пробел.

В языке СИ есть две очень удобные функции puts и gets, позволяющие вводить и выводить строку символов. Пример их использования показан ниже:

```
#include<stdio.h>
main()
{
char q[40]; /*объявление строки символов*/
puts(“Введите строку символов”);
gets(q); /*ввод строки символов*/
puts(q); /*вывод строки символов*/
}
```

В результате работы программы вначале на экране появится текст:

Введите строку символов ,

после чего следует ввести какую-либо строку символов. Эта информация при помощи оператора gets будет присвоена элементам символьного массива q. Оператор puts выведет строку символов.

## Практическая работа 22. Условный оператор

**Цель:** изучить работу условных операторов в C++.

### ХОД РАБОТЫ

**Пример 1.** Напишите программу решения квадратного уравнения с проверкой на наличие вещественных (не комплексных) корней на основе только операторов if. Вид квадратного уравнения:  $ax^2 + bx + c = 0$

Как известно, квадратное уравнение будет иметь вещественные корни, если его дискриминант будет неотрицательным, т.е. когда

$$D = b^2 - 4ac \geq 0$$

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int main(void) {
float a, b, c;
float D, x1, x2, x;
```

```

printf("\n\t Equation a*x^2 + b*x + c = 0\n");
printf("\n\t Enter the coefficient a: ");
scanf_s("%f", &a);
printf("\t Enter the coefficient b: ");
scanf_s("%f", &b);
printf("\t Enter the coefficient c: ");
scanf_s("%f", &c);
D = b*b - 4*a*c;
if (D >= 0 && a != 0) {
x1 = -b/(2*a) + (float)sqrt(D)/(2*a);
x2 = -b/(2*a) - (float)sqrt(D)/(2*a);
printf("\n\t The roots of the equation:\n\t x1 = %1.4f, x2 = %1.4f\n", x1, x2);
}
if (D < 0)
printf("\n\t The roots of complex\n");

if (a == 0 && b != 0) {
x = -c/b;
printf("\n\t As a = %1.0f,\n\t the solution of the equation is: %1.4f\n", a, x); }

printf("\n Press any key: ");
_getch();
return 0;
}

```

В программе последовательно проверяются условия с помощью операторов **if**. В последнем случае, когда коэффициент  $a = 0$ , квадратное уравнение вырождается и превращается в линейное уравнение. Решение в этом случае очевидно.

В программе применены функции `scanf_s()` вместо стандартной функции `scanf()` языка C. Это сделано для того, чтобы по этим функциям не было предупреждений (warning) в MS Visual Studio 2008.

Кроме того, в программу подключена библиотека `math.h` для действий с математическими функциями, например, `sqrt()`.

В первом операторе **if** применено логическое условие **И** ( `&&` ) для проверки того, что дискриминант не равен отрицательному значению и одновременно чтобы первый коэффициент квадратного уравнения не был равен нулю. Аналогичное условие прописано и для последнего оператора **if**.

### Задание 1

1. Объясните включение **float** перед функцией `sqrt()`.
2. В программу введите изменения для вычисления комплексных корней квадратного уравнения без подключения дополнительных библиотек.
3. В программе предусмотрите ситуацию, когда все коэффициенты квадратного уравнения равны нулю.
4. В программе предусмотрите подстановки найденных корней в заданное квадратное уравнение с выводом возможной невязки. Предусмотрите также меры по уменьшению невязки.
5. Предусмотрите циклический ввод коэффициентов квадратного уравнения и вывода решения троекратно.
6. Предусмотрите проверку всех возможных условий задания коэффициентов квадратного уравнения.

**Пример 2.** Напишите программу решения квадратного уравнения с проверкой на наличие вещественных корней на основе конструкции **if-else**. Вид квадратного уравнения:

$$ax^2 + bx + c = 0$$

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

int main(void)
{
    float a, b, c;
    float D, x1, x2, x;
    printf("\n\t Equation a*x^2 + b*x + c = 0\n");
    printf("\n\t Enter the coefficient a: ");
    scanf_s("%f", &a);
    printf("\n\t Enter the coefficient b: ");
    scanf_s("%f", &b);
    printf("\n\t Enter the coefficient c: ");
    scanf_s("%f", &c);
    D = b*b - 4*a*c;
    if (D >= 0 && a != 0 && b != 0) {
        x1 = -b/(2*a) + (float)sqrt(D)/(2*a);
        x2 = -b/(2*a) - (float)sqrt(D)/(2*a);
        printf("\n\t The roots of the equation:\n\t x1 = %1.4f, x2 = %1.4f\n", x1, x2);
    }
    else {
        if (a == 0 && b != 0) {
            x = -c/b;
            if (c != 0)
                printf("\n\t As a = %1.0f,\n\t the solution of the equation is: %1.4f\n", a, x);

            else
                printf("\n\t As a = %1.0f and c = %1.0f,\n\t the solution of the equation is: %1.0f\n", a, -x);
        }
        if (D < 0)
            printf("\n\t The roots of complex\n");
    }

    printf("\n Press any key: ");
    _getch();
    return 0;
}
```

В программе использованы вложенные операторы **if**.

## Задание 2

1. Добавьте в программу преобразование типов при использовании функции `sqrt()`.

2. В программе предусмотрите вычисление комплексных и мнимых корней квадратного уравнения.
3. Предусмотрите ввод целочисленных коэффициентов квадратного уравнения, используя тип **int**. В отчет вставьте возможный результат выполнения программы.
4. Напишите программу ввода вещественных чисел и вывода абсолютного значения этого числа на основе конструкции **if-else**.

**Пример 3.** Напишите программу расчета простого арифметического выражения на основе оператора **switch**.

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>

int main (void) {
    float value1, value2;
    char operat;

    printf("\n\t Printed on the keyboard expression: ");
    scanf_s("%f%c%f", &value1, &operat, sizeof(char), &value2);

    switch (operat) {

    case '+':
        printf("\n\t Result: %1.4f\n", value1 + value2);
        break;

    case '-':
        printf("\n\t Result: %1.4f\n", value1 - value2);
        break;
        case '*':
            printf("\n\t Result: %1.4f\n", value1 * value2);
            break;

    case '/':
        if (value2 == 0.0)
            printf("\n\t Division by zero.\n");
        else
            printf("\n\t Result: %1.4f\n", value1 / value2);
            break;

    default:
        printf("\n\t Unknown arithmetic operator\n\t error or enter a number.
Break!\n");
        break;

    } // End switch

    printf("\n Press any key: ");
    _getch();
    return 0;
}
```

```
}
```

В программе использована полная форма оператора **switch**. Оператор **break** инициирует немедленный выход из оператора **switch**. Возможно использование вложенных операторов **switch**.

### Задание 3

1. Проверьте деление числа на нуль и ввода недопустимого символа.
2. Примените условие равенства нулю вводимого числа без знака "==".
3. Напишите программу расчета простого выражения с помощью конструкций **if-else if-else**.
4. Напишите программу деления суток на "Morning" (утро), "Day" (день), "Afternoon" (послеобеденное время), "Evening" (вечер), "Night" (ночь). Время ввода задается пользователем с клавиатуры.

**Пример 4.** Напишите программу вычисления двух целых случайных чисел и определения наибольшего из них. Определение наибольшего числа произведите с помощью оператора условия ?

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h> // Для функций случайных чисел
#include <time.h>

int main (void) {
int a, b, maxab;
unsigned int some;
long int L;

L = (long) time(NULL);
some = (unsigned) L/2;
srand(some);
a = rand();
b = rand();
printf("\n\t Random numbers: a = %d; b = %d\n", a, b);
// Оператор условия для определения максимального числа
maxab = (a > b) ? a : b;

printf("\n\t Maximum number: %d\n", maxab);

    printf("\n Press any key: ");
    _getch();
    return 0;
}
```

В программе использованы функции генерации псевдослучайных чисел `rand()` и задания исходного псевдослучайного числа `srand()`. Указанные функции входят в стандартную библиотечную функцию `stdlib.h`. Функция `time()` входит в библиотечную функцию `time.h`, которая поддерживает функции, обращающиеся к системному времени.

Для переменных `L` и `some` выполнено приведение типов.

При каждом обращении к функции `rand()` возвращается целое в интервале между нулем и значением `RAND_MAX`, которое в любой реализации должно быть не меньше числа 32767

#### Задание 4

1. При выводе максимального числа предусмотрите сообщение об имени числа, т.е. **a** или **b**.
2. В цикле сформируйте вектор десяти случайных чисел из интервала  $[0;1]$  и выведите на дисплей.
3. Для задания системного времени примените директиву **define** `N X`, где **X** – номер вашего варианта (№ варианта – это номер по журналу).
4. Примените оператор условия `?` для определения абсолютного значения вещественного числа, которое должно вводиться пользователем с клавиатуры.

**Пример 5.** Используя оператор условия `?` и переключатель **switch** напишите программу определения времени года по вводимым числам от 1 до 12, считая, что цифра 1 соответствует январю, цифра 2 – февралю и т.д.

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>

int main (void) {
    int x;
    printf("\n\t Enter a whole number between 1 and 12: ");
    scanf_s("%d", &x);

    switch (x > 0 && x < 3 ? 1 :
           x == 12 ? 1 :
           x > 2 && x < 6 ? 2 :
           x > 5 && x < 9 ? 3 :
           x > 8 && x < 12 ? 4 :
           x > 12 || x < 1 ? 5 : 5)
    {
    case 1 :
        printf("\n\t This Winter\n");
        break;
    case 2 :
        printf("\n\t This Spring\n");
        break;
    case 3 :
        printf("\n\t This Summer\n");
        break;
    case 4 :
        printf("\n\t This Autumn\n");
        break;
    case 5 :
        printf("\n\t This is a mistake (Error)\n");
    }
}
```

```

    printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В программе три месяца зимы кодируются цифрой 1, три месяца весны – цифрой 2, три месяца лета – цифрой 3, три месяца осени – цифрой 4. Если введенная цифра не входит в целочисленный интервал [1; 12], то эта ситуация кодируется цифрой 5.

### Задание 5

1. Проверьте программу по вводимым вещественным числам. Объясните результат.
2. Проверьте программу по вводимым буквам или знакам, имеющихся на клавиатуре. Объясните результат.
3. В программе вместо 5-го пункта, т.е. вместо **case 5**, примените операцию **default**.
4. Напишите программу определения времени года по вводимым числам без операторов условия ?.

## Практическая работа 23. Оператор цикла (часть 1)

**Цель:** изучить виды циклов и работу с ними в C++.

Рассмотрим примеры программ с операторами циклов **while**, **for** и **do-while**.

**Пример 1.** Напишите программу вывода на экран пользователя целых положительных чисел с помощью оператора **while**. Начальное и последнее число должно задаваться пользователем с клавиатуры.

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>

int main(void)
{
    int i, j = 0, n;
    printf("\n\t Enter the primary natural number: ");
    scanf_s("%d", &i);
    printf("\t Enter the last natural number: ");
    scanf_s("%d", &n);

    printf("\n\t The numbers are:");
    while (i <= n) {
        printf("\n\t %3d", i);
        ++i;
        ++j;
    }
    printf("\n\t Total numbers: %d\n", j);

    printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В программе использована функция `scanf_s()`, принятая в MS Visual Studio. В программе применено инкрементирование переменных, принятое в языке C, а именно `++i` или `++j` означает, что переменные увеличиваются на единицу. При этом знаки `++` могут располагаться перед именем переменной или после. Отличие в том, что `++i` – это значение переменной после увеличения, а `i++` – сначала переменная имеет заданное значение, а потом происходит ее увеличение. Для переменных цикла обе формы равнозначны.

Условием цикла является то, что пока переменная `i` меньше или равна переменной `n` (предполагается, что `n` больше начального значения `i`), то будут выполняться действия (печать и увеличение переменной `j`), заложенные в теле цикла. Расчет выражения, заключенного в круглые скобки оператора, предназначен для проверки нестроого неравенства переменной `i` по отношению к переменной `n`. Если это неравенство выполняется, то в теле цикла происходят печать и увеличение (инкрементирование) переменных `i, j`.

### Задание 1

1. Предусмотрите ввод только отрицательных чисел.
2. Предусмотрите ввод только неотрицательных чисел.
3. Предусмотрите вывод чисел и их порядковые номера (т.е. в два столбца).
4. Предусмотрите вывод чисел на консоль в виде строки.
5. Предусмотрите ввод и вывод только вещественных чисел.
6. С учетом предыдущего пункта предусмотрите вывод чисел и их порядковые номера (т.е. в два столбца).
7. Рассмотреть варианты форм инкрементирования. Выполните вывод на консоль.

**Пример 2.** Напишите программу расчета значений функции распределения потока Эрланга 4-го порядка с параметром  $\lambda = 2.3$  на отрезке времени от 0 до 5 (условных единиц) с шагом 0.2:

$$F_k(t) = 1 - \sum_{j=0}^k \frac{(\lambda t)^j}{j!} e^{-\lambda t}$$

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

int main(void)
{
    int j, f, k = 4;
    double F, Lt, s, t = 0.0, Tend = 5.0;
    float L = 2.3F;

    printf("\n Erlang function of order %d, Lambda = % 1.2f:\n\n", k, L);

    for ( ; t <= Tend; t += 0.2) {
        s = 0.0; f = 1; Lt = 1.0;
        for (j = 1; j <= k; ++j) {
            f *= j;
            Lt *= (L*t);
            s += (Lt/f)*exp(-L*t);
        }
    }
}
```



```

}
F = 1 - exp(-L*t) - s;
printf("\t %lg\n", F);
}

    printf("\n\n Press any key: ");
    _getch();
    return 0;
}

```

В программе использованы вложенные циклы **for**. Применены специфические условия изменения переменных (но характерных для языка **C**):

```

t += 0.2;    /* t = t + 0.2; */
    f *= j;    /* f = f * j; */
    Lt *= (L*t); /* Lt = Lt * (L*t); */

```

Вычисление факториала выполнено с помощью произведения и выделения отдельного слагаемого суммы, когда рассчитывается 0!

В программе также предусмотрено накопление суммы для каждого шага времени *t*. Для расчета экспоненты в заголовке программы включена библиотека математических функций **#include <math.h>**.

## Задание 2

1. В заголовке внешнего цикла **for** добавьте первое поле и уберите третье поле.
2. Предусмотрите вывод количества значений рассчитанной функции Эрланга.
3. Предусмотрите ввод параметров программы, а именно: порядок потока Эрланга *k*, параметр  $\lambda$ , конечную величину временного интервала, шаг по времени.

**Пример 3.** Вычислите с точностью до "машинного нуля" значение суммы числового ряда:

$$\frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{2 \cdot 3 \cdot 4} + \frac{1}{3 \cdot 4 \cdot 5} + \frac{1}{4 \cdot 5 \cdot 6} + \dots$$

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>

int main(void)
{
    double denom;
    double sum1 = 0.0, sum2 = 0.0;
    int k = 1;
    denom = k * (k + 1) * (k + 2); // знаменатель ряда

    do {
        sum1 = sum2;
        sum2 += 1.0 / denom;
        denom = denom / k * (k + 3);
        ++k;
    } while (sum1 < sum2);

    printf("\n\t The amount of numerical series: %lg\n", sum2);
}

```

```

    printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В приведенной программе сумма вычисляется как значение переменной `sum2`. Ее предыдущее значение сохраняется в переменной `sum1`. Так как приближенное значение с добавлением неотрицательных слагаемых не уменьшается, условием продолжения цикла служит отношение `sum1 < sum2` (поскольку растет знаменатель `denom`). Когда при добавлении очередного слагаемого значение суммы остается неизменным (за счет конечной разрядной сетки для представления вещественных чисел), нарушается условие `sum1 < sum2` и цикл прекращается. Таким образом, конечность разрядной сетки представления вещественных чисел в компьютере определяет собой "машинный нуль". Инициализация знаменателя сделана до начала цикла. Форматный вывод результата выполнен с помощью спецификатора символа "l".

## Практическая работа 24. Оператор цикла (часть2)

**Цель:** изучить виды циклов и работу с ними в C++.

### ХОД РАБОТЫ

#### Задание 1

Подсчитайте число итераций цикла для подсчета суммы ряда.

Напишите программу с нулевой инициализацией переменной `k`.

Напишите условие "загрубения" результата подсчета суммы ряда.

В программе примените тип **float** вместо типа **double**. Проанализируйте результат.

Вместо оператора цикла **do-while** примените иной оператор цикла.

**Пример 1.** Произведите реверс цифр заданного целого числа, вводимого с клавиатуры пользователем.

Задача заключается в том, чтобы, например, число 123 переписать как 321.

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>

```

```

int main(void)

```

```

{
    long int x, r;

```

```

    printf("\n Enter an integer: ");
    scanf_s("%ld", &x);

```

```

    printf("\n Reverse-digit number %ld:\n\n\t", x);

```

```

    do {
        r = x % 10;
        printf(" %ld", r);
        x = x / 10;
    } while ( x != 0);

```

```

    printf("\n\n Press any key: ");

```

```

_getch();
return 0;
}

```

В программе применена операция арифметическая операция деления по модулю, которая имеет символ процента, т.е. "%". Любой остаток, получающийся в результате деления целых чисел, будет отброшен. В шкале старшинства оператор деления по модулю имеет приоритет, равный приоритету операторов умножения и деления. Переменные, используемые в программе, объявлены как длинные числа, поэтому применен тип `long int` (или `long`). В некоторых компиляторах имеются отличия между типами `int` и `long int` в смысле максимально поддерживаемого значения числа.

## Задание 2

Предусмотрите подсчет итераций заданного цикла.

Напишите программу по реверсу числа с оператором цикла **while**. Протестируйте обе программы, в том числе по вводу числа 0.

Определите максимальное число вида 123...987..., для которого еще можно применить тип **long int**.

**Пример 2.** Напишите программу распечатки четных целых чисел от 0 до 30.

```

#include <stdio.h>
#include <conio.h>

int main (void)
{
    int x;
    printf("\n\t Even numbers from 0 to 30:\n\n");

    for (x = 0; x < 31; x++) {
        if ( x % 2 ) continue;
        printf("\t\t %3d\n", x);
    }

    printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В программе в качестве проверки условия использовано деление по модулю ( $x\%2$ ). Если остаток от деления числа  $x$  не равен нулю, то утверждение (оператор, инструкция) **continue** передает управление непосредственно инструкции, проверяющей условное выражение, после чего циклический процесс продолжается. С помощью программы выводятся только четные числа, а при обнаружении нечетного числа происходит преждевременный переход к следующей итерации цикла, и функция **printf()** опускается. Функция **printf()** включена в тело цикла оператора **for**.

## Задание 3

В программу включите действия подсчета суммы четных чисел.

Подсчитайте число итераций оператора цикла.

В программе вместо цикла **for** примените цикл **while**. Объясните действие инструкции **continue**.

В программе вместо цикла **for** примените цикл **do-while**. Объясните действие инструкции **continue**.

Напишите программу вывода четных чисел без оператора **continue**. Подсчитайте число итераций оператора цикла.

Сделайте вывод четных чисел из интервала от  $X$  до  $10 \cdot X$ , где  $X$  – номер вашего варианта (№ варианта – это номер по журналу). Подсчитайте сумму четных чисел. Измените условие для вывода четных чисел.

**Пример 3.** Напишите программу подсчета суммы трех чисел из трех вложенных циклов и, если сумма делится без остатка на число 3, то прекратите сравнение чисел и выйдите из циклов с последующей распечаткой этой суммы и слагаемых. Первый цикл начинается с 51 до 1, второй цикл – с 41 до 1, третий цикл – с 1 до 50. Первый цикл изменяется через 7 единиц, второй цикл изменяется на 1, третий цикл изменяется на 7 единиц.

Программный код решения примера с оператором **goto**:

```
#include <stdio.h>
#include <conio.h>

int main (void) {
    int a, b, c, i, j, k;
    a = b = c = 0;

    for (i = 51; i >= 1; i -= 7 )
    for (j = 41; j >= 1; --j )
    for (k = 1; k <= 50; k += 7)
    if ( ((i + j + k) % 3 == 0) && (i < 51) && (j < 41))
    {a = i; b = j; c = k;
        goto sum3;
    }
sum3:
printf("\n\t The sum (%d?+?%d?+?%d) is equal %d\n", a, b, c,
    a + b + c);

printf("\n Press any key: ");
_getch();
return 0; }
```

В программе использовано декрементирование (уменьшение на единицу) переменной  $j$  в форме  $--j$ . Переменная  $i$  с каждой итерацией цикла уменьшается на 7 единиц. Переменная  $k$  с каждой итерацией цикла увеличивается на 7 единиц. После оператора **if** включены фигурные скобки для выполнения нескольких действий при выполнении заданного условия оператора **if**.

**Примечание.** Оператор **goto** нельзя применять для перехода в тело цикла, т.е. метка не должна быть внутри оператора цикла. Метка может появиться текстуально до или после оператора **goto**.

#### Задание 4

В программу включите подсчет числа итераций каждого из циклов.

Вместо оператора безусловного перехода **goto** примените оператор **break**. Известно также подсчет числа итераций каждого из циклов. Сравните с аналогичными результатами предыдущего пункта задания.

В программе вместо операторов **for** примените операторы **while**. Выполните действия двух предыдущих пунктов задания.

Напишите программу ввода символа до тех пор, пока не будет введен заранее определенный символ, например 'w'. Используйте оператор **goto**. Предусмотрите отступ от левого края дисплея.

**Пример 4.** Напишите программу распечатки на консоль простых чисел из диапазона от 2 до N, где N – число, вводимое пользователем с клавиатуры, которое не превосходит, например, 1000.

Как известно, простое число – это целое положительное число больше единицы, которое не делится без остатка ни на одно другое целое положительное число, кроме единицы и самого себя. Единица не считается простым числом.

Возможный программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h> // для exit()

#define Nmax 50

int main (void) {
    int i, j;
    int ok, in;
    int N;

    printf("\n Enter an integer from 2 to %d: ", Nmax);
    in = scanf_s("%d", &N);

    if (in == 0 || N < 2 || N > Nmax)
    {
        printf("\n Error input. Press any key: ");
        _getch();
        exit(1);
    }

    printf("\n Prime numbers from 2 to %d:\n\n", N);

    for ( i = 2; i <= N; i++ )
    {
        ok = 1;
        for ( j = 2; j < i; j++ )
            if ( (i%j) == 0 )
                ok = 0;
        if ( ok )
            printf(" %3d", i);
    }

    printf("\n\n Press any key: ");
    _getch();
    return 0;
}
```

В программе с помощью оператора **if** осуществляется проверка правильности ввода данных с клавиатуры. Кроме того, этим же оператором проверяется остаток от деления двух чисел и условной истинности, когда переменная *ok* не равна нулю. С помощью препроцессорной директивы **define** определяется верхняя допустимая граница для простых чисел.

## Задание 5

В программе вместо оператора цикла **for** примените оператор **while**.

В программе примите допустимое число **Nmax**, равное 9999. Предусмотрите при этом форматированный вывод на консоль простых чисел построчно, по 15 чисел в каждой строке. Используйте тернарный оператор **?:**.

Предыдущий пункт задания выполните с помощью операторов **if**, **else**.

## Практическая работа №25. Программирование микроконтроллера на языке C

**Цель:** изучить команды программирования микроконтроллеров

### ХОД РАБОТЫ

Язык программирования Си – высокоуровневый язык, основной идеей которого является работа с функциями, аргументы которых передаются как параметры. Любая функция состоит из имени и скобок с аргументами. Аргументы разделяются запятыми. Основное тело программы заключено в функцию `main()`. Любая программа должна содержать эту функцию т.к. с неё начинается работа программы.

Рассмотрим пример простой программы для микроконтроллера AVR для компилятора WinAVR:

```
{codecitation class="brush: vb; gutter: true;" width="700px" }
#include <avr/io.h>
#define F_CPU = 1000000UL;
#include <util/delay.h>
int main(void){
DDR0 = 0b00000100; // второй бит порта D – выход, остальные входы
while(1){
PORTD ^= 0b00000100; // исключающее или – сменить состояние второго бита
_delay_ms(250); // ожидание 250 мс
}
return 0;
}
{/codecitation }
```

В этой программе содержатся основные элементы языка.

Все препроцессорные команды начинаются со знака “#”.

```
{codecitation class="brush: vb; gutter: true;" width="700px" }
```

`#include <avr/io.h>` – препроцессорная команда которая подключает библиотеку аппаратных описаний микроконтроллера. В ней содержатся аппаратные адреса портов и регистров микроконтроллера. Используя эту библиотеку можно использовать легкозапоминающиеся мнемоники такие как: `DDR0`, `PORTD`, `TCNT0`, `MCUCR`... а не их адреса, которые не так легко запомнить. Например, ресли рассмотреть файл `io.h` для микроконтроллера ATtiny13:

```
...
/* Timer/Counter0 (8-bit) */
#define TCNT0          _SFR_IO8(0x32)
/* Timer/Counter 0 Control Register B */
#define TCCR0B         _SFR_IO8(0x33)
# define FOC0A         7
# define FOC0B         6
# define WGM02         3
# define CS02          2
# define CS01          1
```

```
# define CS00          0
```

```
...
```

```
{/codecitation }
```

Заметить, что адресу регистра Timer0 присваивается легкозапоминаемая мнемоника TCNT0. Поэтому в своей программе мы можем писать так:

```
{codecitation class="brush: vb; gutter: true;" width="700px" }
```

```
TCCR0B |= (0 << CS02) | (1 << CS01) | (1 << CS00); // устанавливаем коэффициент  
// деления тактовой частоты
```

```
TCNT0 = 239; // фактически записываем значение в регистр с адресом 0x32
```

```
DDRB |= (1 << 1) | (1 << 2); // Делаем выходами первый и второй бит порта В (он же  
регистр ввода/вывода с адресом 0x17)
```

```
{/codecitation }
```

Если файл \*.h заключен в двойные кавычки, то поиск файла проводится в папке проекта.

```
{codecitation class="brush: vb; gutter: true;" width="700px" }
```

```
#define F_CPU = 1000000UL; // указывает компилятору, что наш кристалл такти-  
руется от резонатора с тактовой частотой в миллион герц т.е. 1МГц. "UL" – после  
числа показывает, что число Unsigned Long т.е. беззнаковое типа "long"
```

```
{/codecitation }
```

```
{codecitation class="brush: vb; gutter: true;" width="700px" }
```

int main(void) – означает, что функция main возвращает значение типа int т.е. значения от 0 до 65535.

```
{/codecitation }
```

Открывающаяся фигурная скобка "{" показывает начало исполняемой части функции.

```
{codecitation class="brush: vb; gutter: true;" width="700px" }
```

while(1){...} – бесконечный цикл, код заключенный в фигурные скобки будет выполняться бесконечно.

```
{/codecitation }
```

"/" – символы начала комментария, текст в строке после этих символов не включается в программу при компиляции.

"Return 0" – возвращает значение функции main(). Все функции должны заканчиваться закрывающимися фигурными скобками "}". После каждой команды надо ставить ";;".

Основные типы данных табл.:

Таблица 23

| Тип           | Название           | Размер в битах | Диапазон значений                    |
|---------------|--------------------|----------------|--------------------------------------|
| char          | Символ             | 8              | -128...127                           |
| int           | Целое число        | 16             | -32768...32767                       |
| float         | Вещественное число | 32             | $\pm 10^{-38} \dots \pm 3 * 10^{38}$ |
| long int      | Длинное целое      | 32             | $\pm 2147483647$                     |
| unsigned char | ASCII символ       | 8              | 0...255                              |
| unsigned int  | Беззнаковое целое  | 16             | 0...65535                            |

## Практическая работа №26. Программирование микроконтроллеров на языке C

**Цель:** изучить команды программирования микроконтроллеров

### ХОД РАБОТЫ

**Задание 1.** Найти ошибку в программе:

**Пример 1.**

```
#include <iostream.h>
void main()
{int a=1, b=2, c;
c=sum(a,b);
cout<<c;
}
int sum(int x,int y)
{return x+y;}
```

**Задание 2.** Определить результат выполнения программы:

**Пример 2.**

```
#include <iostream.h>
void mul(int,int);
int S;
void main()
{int a=2,b=3;
mul(a,b); a=2*S; mul(a,b);
cout<<S;
}
void mul(int x,int y)
{S=x*y; }
```

**Пример 3.** Напишите программу вычисления квадратного корня числа по методу Ньютона–Рафсона с использованием функции расчета квадратного корня числа и функции расчета абсолютного значения числа.

При вычислении квадратного корня из числа следует помнить, что подкоренное выражение не должно быть отрицательным.

Алгоритм метода Ньютона–Рафсона для вычисления квадратного корня числа:

*Шаг 1.* Выбрать приближительное значение 1.

*Шаг 2.* Если  $|\text{guess}^2 - x| < \epsilon$ , перейти к шагу 4.

*Шаг 3.* Установить приближительное значение, равное  $(x/\text{guess} + \text{guess})/2$ , и перейти к шагу 2.

*Шаг 4.* Считать приближительное значение квадратным корнем числа.

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>

// Функция абсолютного значения числа
double absValue(double x)
```



```

{
    if (x < 0) x = -x;
    return (x);
}

// Функция расчета квадратного корня из числа
double squareRoot (double x) {
const double epsilon = 0.000001;
double guess = 1.0; //Начальное приближение

while (absValue(guess*guess - x) >= epsilon)
guess = (x/guess + guess)/2.0;

return (guess);
}

int main (void)
{
    double result, X;

    printf("\n\t Enter a number: ");
    scanf_s("%lf", &X);

// Обращение к функции с фактическим параметром
    result = squareRoot(X);
printf("\n\t Square root of \"%1.4f\" is: %1.8f\n", X, result);

    printf("\n Press any key: ");
    _getch();
    return 0;
}

```

В программе используются три функции: main(), absValue(), squareRoot().

Вспомогательные функции расположены в определенном порядке: сначала функция absValue(), а потом функция squareRoot(). В функцию absValue() передается значение **guess**, которое вычисляется в функции squareRoot(), находящейся в условии оператора цикла while. Когда условие для оператора цикла будет ложным, т.е. когда значение корня будет меньше заданного числа epsilon, то полученное значение возвращается в вызывающую функцию main().

В программе использована переменная **epsilon**, определенная с помощью спецификатора **const**, что делает переменную неизменной. При таком объявлении компилятор определяет ее как константное значение. Таким переменным нельзя присваивать значения в программе, нельзя их инкрементировать или декрементировать.

Возможный результат выполнения программы показан на **рис. 1**.

```

Enter a number: 2
Square root of "2.0000" is: 1.41421356
Press any key: _

```

**Рис. 1.** Результат итерационного извлечения квадратного корня из двух

### Задание 3

1. В качестве формальных параметров функций возьмите начальные буквы своей фамилии, имени из двух букв. В качестве возвращаемого значения из функции `squareRoot()` – месяц своего рождения.
2. В функции `absValue()` примените оператор условия `?` вместо оператора `if` без определения дополнительной переменной.
3. В главной функции `main()` введите проверку ввода неотрицательного числа.
4. В программу введите прототипы функций.
5. Зафиксируйте результат вычисления квадратного корня от числа  $X$ , где  $X$  – номер вашего варианта (№ варианта – это номер по журналу).
6. В качестве начального приближения возьмите  $1.5 * X$ , где  $X$  – номер вашего варианта (№ варианта – это номер по журналу).
7. В программе примените тип **float** вместо типа **double**.

Программный код решения примера

```
#include <conio.h>
#include <string.h>
#include <stdarg.h>
#include <float.h>

/* Прототип функции с переменным числом аргументов*/
double varfun(char str[], double v1, ...);
int main(void)
{
    double v1 = 10.0,
           v2 = 2.5,
           control;
    char str[121];

    printf("\n Enter one of the key words 'mean' or 'sum': ");
    gets_s(str, 120);

    control = varfun(str, v1, v2, 7.5, 0.0);
    if (control < DBL_MAX)
        printf("\n Result: %lf\n", control);

    else
        printf("\n Invalid input keyword.\n");

    printf("\n\n ... Press any key: ");
    _getch();
    return 0;
}

/* Определение функции с переменным числом аргументов */
double varfun( char str[], double v1, ...) {
/* Указатель на переменные списка аргументов */
    va_list parg;
```

```

double sum = v1;
double value = 0.0; /* значение аргумента */
int count = 1; /* начальное количество аргументов */
int k, p;
char *ch = "mean";
char *ch2 = "sum";

// Лексиграфическое сравнение строк
k = strcmp(str, ch);
if ( !k ) // k == 0
    p = 1;

else if ( k ) // k != 0
{
    k = strcmp(str, ch2);
    if ( !k )
        p = 2;

    else
        p = 0;
}
if ( p == 1 )
{
    va_start(parg,v1); /* инициализация указателя parg */
    /* Просмотр списка аргументов*/
    while( (value = va_arg(parg, double)) != 0.0)
    {
        // Суммирование числовых аргументов функции varfun()
        sum += value;
        count++;
    }

    /* Завершение процесса считывания аргументов */
    va_end(parg);
    return sum/count;
}

if ( p == 2 )
{
    va_start(parg,v1);

    while( (value = va_arg(parg, double)) != 0.0)
    {
        sum += value;
        count++;
    }
    va_end(parg);
    return sum;
}
return DBL_MAX;
}

```

Сохраните результат, проверьте работу микроконтроллера.

## Литература

### Основные источники:

1. Бурьков, Д. В. Применение IT-технологий в электроэнергетике: Mathcad, Matlab (Simulink), NI Multisim: учебное пособие / Д. В. Бурьков, Н. К. Полуянович; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2018. - 126 с. - ISBN 978-5-9275-3086-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1088095> (дата обращения: 3.06.2020).
2. Дорогов, В. Г. Основы программирования на языке C: учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2020. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0809-9. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1082440> (дата обращения: 3.06.2020)
3. Молоканова, Н. П. Автоматическое управление. Курс лекций с решением задач и лабораторных работ: учебное пособие / Н. П. Молоканова. - Москва: Форум, 2021. - 224 с.: ил. - (Профессиональное образование). - ISBN 978-5-91134-593-8. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1160864> (дата обращения: 3.06.2020)

### Дополнительные источники

1. Вепринцев, В. И. Автоматизированный лабораторный практикум с удаленным доступом для исследования электрических цепей [Электронный ресурс] : учеб. пособие / В. И. Вепринцев, А. С. Глинченко, В. И. Коваленок, В. А. Комаров. - Красноярск: Сиб. федер. ун-т, 2011. - 188 с. - ISBN 978-5-7638-2340-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/441861> (дата обращения: 3.06.2020)
2. Кузин, А. В. Программирование на языке Си: учебное пособие / А.В. Кузин, Е.В. Чумакова. — Москва: ФОРУМ: ИНФРА-М, 2020. — 143 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-556-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/961653> (дата обращения: 3.06.2020)
3. Маркелов, С. Н. Электротехника и электроника: учебное пособие / С.Н. Маркелов, Б.Я. Сазанов. — Москва: ИНФРА-М, 2021. — 267 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014453-5. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1190677> (дата обращения: 3.06.2020)

### Интернет-ресурсы:

1. Информационный портал Mathcad: официальный сайт. – URL: <https://www.ptc.com/en/products/mathcad> (дата обращения: 03.06.2020). – Текст: электронный.
2. Информационный портал NI Multisim: официальный сайт. – URL: <http://sine.ni.com/nips/cds/view/p/lang/ru/nid/201800> (дата обращения: 03.06.2020). – Текст: электронный.
3. Информационный портал микроконтроллеров PIC и AVR: официальный сайт. – URL: <http://www.microchip.com> (дата обращения: 03.06.2020). – Текст: электронный.