

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ДИСЦИПЛИНЕ
«МИКРОПРОЦЕССОРНАЯ ТЕХНИКА»**

Выпускная квалификационная работа специалиста
по специальности 050501 Профессиональное обучение
специализации «Вычислительная техника»

Идентификационный код ВКР: 106

Екатеринбург 2016

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующий кафедрой ИС

_____ Н. С. Толстова

« ____ » _____ 2016 г.

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ДИСЦИПЛИНЕ
«МИКРОПРОЦЕССОРНАЯ ТЕХНИКА»

Выпускная квалификационная работа специалиста
по специальности 050501 Профессиональное обучение
специализации «Вычислительная техника»

Идентификационный код ВКР: 106

Исполнитель:

студент группы ВТ-503

К. С. Михалев

Руководитель:

ст. преподаватель

В. В. Мешков

Нормоконтролер:

Т. В. Рыжкова

РЕФЕРАТ

Выпускная квалификационная работа содержит 82 листа машинописного текста, 11 иллюстраций, 30 литературных источников, 1 приложение.

Ключевые слова: ЛАБОРАТОРНАЯ РАБОТА, УЧЕБНЫЙ ПЛАН, РАБОЧАЯ ПРОГРАММА, МЕТОДИЧЕСКИЕ УКАЗАНИЯ, ТРЕНАЖЕР, ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ.

Цель работы – разработать комплекс лабораторных работ для работы с программной моделью микропроцессорного стенда УМПК-80.

Для достижения поставленной цели были решены следующие *задачи*:

- проанализирована учебно-методическая документация подготовки студентов по дисциплине «Микропроцессорная техника»;
- разработан комплекс лабораторных работ;
- разработаны методические указания для проведения лабораторных работ.

Результаты работы могут использоваться для обучения студентов заочной и очной форм обучения по дисциплине «Микропроцессорная техника» на кафедре информационных систем и технологий РГППУ.

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ учебно-методической документации	6
1.1 Государственный образовательный стандарт	6
1.2 Учебный план	11
1.3 Рабочая программа.....	13
1.4 Методические указания для выполнения лабораторных работ	16
2 Разработка лабораторных работ	18
2.1 Понятие лабораторной работы	18
2.2 Методика проведения лабораторных работ	21
2.3 Разработка тем лабораторных работ	24
2.4 Лабораторный практикум	27
2.4.1 Лабораторная работа № 1 «Изучение интерфейса и режима работы программной модели стенда УМПК-80».....	27
2.4.2 Лабораторная работа № 2 «Изучение выполнения команд загрузки и перемещения данных».....	47
2.4.3 Лабораторная работа № 3 «Изучение выполнения команд ввода и вывода информации».....	55
Заключение	77
Список использованных источников информации	78
Приложение Лист задания	Ошибка! Закладка не определена.

ВВЕДЕНИЕ

В настоящее время широкое применение микропроцессорных систем (МС) привело к существенному изменению характера деятельности разработчиков и ремонтников разнообразной радиотехнической аппаратуры и оборудования. Благодаря МС проектирование и ремонт радиотехнических схем уже не является приоритетной задачей. На главное место выходит умение грамотно написать и отладить программное обеспечение для разработанной и находящийся на обслуживании радиотехнической аппаратуры и оборудования.

Ввиду сложности выполняемых работ по проектированию и обслуживанию оборудования на базе МС, для студентов учебных заведений, в программе обучения которых есть дисциплины, связанные с МС, требуются хорошо сформированное сочетание теоретической и практической подготовки. Это сочетание позволяет с помощью практических занятий закрепить полученные теоретические сведения. Такой комплексный подход даёт возможность подготовить специалистов способных эффективно справляться с задачами разработки, использования и обслуживания радиотехнической аппаратуры и оборудования на базе МС.

Одна из дисциплин, которая помогает сформировать знания и умения необходимые для подготовки таких специалистов, является: «Микропроцессорная техника».

На кафедре информационных систем и технологий РГППУ по дисциплине «Микропроцессорная техника», для формирования знаний и умений используется методические разработки лабораторных работ на базе стенда УМПК-80. Изучение практики образовательного процесса по дисциплине «Микропроцессорная техника» показывает, что недостаточно внедряются и

используются разнообразные методы обучения, имеет место слабая методическая разработанность их применения при преподавании.

Из вышесказанного можем назвать объект и предмет и цель работы.

Объект работы – подготовка студентов по дисциплине «Микропроцессорная техника».

Предмет работы – формирование системы знаний и умений по дисциплине «Микропроцессорная техника»

Цель работы состоит в том, чтобы разработать комплекс лабораторных работ для работы с программной моделью микропроцессорного стенда УМПК-80.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать учебно-методическую документацию подготовки студентов по дисциплине «Микропроцессорная техника»;
- разработать комплекс лабораторных работ;
- разработать методические указания для проведения лабораторных работ.

1 АНАЛИЗ УЧЕБНО-МЕТОДИЧЕСКОЙ ДОКУМЕНТАЦИИ

Анализ учебно-методической документации включает:

- анализ учебного плана;
- анализ учебной программы;
- анализ методических указаний для выполнения лабораторных работ.

Цель – анализ моделей обучения специалистов и содержания учебных программ, способствующих выработке профессиональных умений.

Метод – анализ учебно-программной документации.

Методическое обеспечение: сборники учебно-программной документации, программы учебных предметов.

1.1 Государственный образовательный стандарт

Анализ учебно-методической документации начинаем с рассмотрения государственного образовательного стандарта высшего профессионального образования.

Первый раздел стандарта: характеристика направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля подготовки «Информационные технологии» профилизации «Компьютерные технологии автоматизации и управления».

Объектами профессиональной деятельности педагогов профессионального обучения выступают участники и средства осуществления целостного педагогического процесса.

Педагог профессионального обучения организует и проводит теоретическое обучение по общепрофессиональным и специальным учебным пред-

метам, а также производственное (практическое) обучение по группам родственных профессий. Организует и проводит учебно-воспитательную работу.

Педагог профессионального обучения должен знать:

Конституцию Российской Федерации; законы Российской Федерации, решения Правительства Российской Федерации и органов управления образованием по вопросам профессионального образования; Конвенцию о правах ребенка; основы трудового законодательства; государственные образовательные стандарты начального профессионального образования; способы проектирования содержания образования при сочетании теоретического и практического обучения; способы отбора адекватных форм, методов и средств обучения; общую и профессиональную педагогику; психологию профессионального образования; многоуровневую систему профессионального образования, квалификационную структуру профессионального образования, его функционирование, закономерности и механизмы; информационные технологии, компьютерную и вычислительную технику.

Второй раздел стандарта: требования к уровню подготовки абитуриента.

Предшествующий уровень образования абитуриента – среднее (полное) общее образование.

Абитуриент должен иметь документ государственного образца о среднем (полном) общем образовании или среднем профессиональном образовании, или начальном профессиональном образовании, если в нем есть запись о получении предьявителем среднего (полного) общего образования, или высшем профессиональном образовании.

Третий раздел стандарта: общие требования к основной образовательной программе направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля подготовки «Информационные технологии» профилизации «Компьютерные технологии автоматизации и управления».

Она состоит из дисциплин федерального компонента, дисциплин национально-регионального компонента, дисциплин по выбору студента, а также факультативных дисциплин.

Программа подготовки педагога профессионального обучения должна предусматривать изучение студентом следующих циклов дисциплин и итоговую государственную аттестацию:

- цикл ГСЭ – общие гуманитарные и социально-экономические дисциплины;
- цикл ЕН – общие математические и естественнонаучные дисциплины;
- цикл ОПД – общепрофессиональные дисциплины;
- цикл ОД – дисциплины отраслевой подготовки;
- ФТД – факультативы.

Содержание национально-регионального компонента основной образовательной программы должно обеспечивать подготовку выпускника в соответствии с квалификационной характеристикой, установленной настоящим государственным образовательным стандартом.

Четвертый раздел стандарта: требования к обязательному минимуму содержания основной образовательной программы направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля подготовки «Информационные технологии» профилизации «Компьютерные технологии автоматизации и управления».

Раздел содержит наименование дисциплин и их основные разделы, индексы и количество часов.

Микропроцессорная техника входит в дисциплины специализации и составляют 180 часа учебного времени.

Пятый раздел стандарта: сроки освоения основной образовательной программы.

Срок освоения основной образовательной программы подготовки педагога профессионального обучения при очной форме обучения составляет 260 недель, в том числе:

Теоретическое обучение, включая научно–исследовательскую работу студентов, практикумы, в том числе лабораторные, а также экзаменационные сессии – 176 недель, практики – не менее 28 недель.

При заочной форме обучения студенту должна быть обеспечена возможность занятий с преподавателем в объеме не менее 180 часов в год.

Шестой раздел стандарта: требования к разработке и условиям реализации основной образовательной программы.

В шестом разделе представлены следующие требования:

- к разработке основной образовательной программы подготовки педагога профессионального обучения;
- к кадровому обеспечению учебного процесса;
- к учебно-методическому обеспечению учебного процесса;
- к материально-техническому обеспечению учебного процесса;
- к организации практик.

Седьмой раздел стандарта: требования к уровню подготовки выпускника направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля подготовки «Информационные технологии» профилизации «Компьютерные технологии автоматизации и управления».

В седьмом разделе представлены требования к уровню подготовки выпускника.

Выпускник высшего профессионального учебного заведения должен уметь решать задачи, соответствующие его квалификации.

Педагог профессионального обучения должен:

иметь представление:

- об индивидуальном и коллективном поведении объектов в природе;

– о локальных, системных, приборных интерфейсах и интерфейсах периферийных устройств и т.д.

быть способен:

– проектировать мультимедийные комплексы, предназначенные для использования в учебном процессе, на основе существующих типовых средств вычислительной техники;

– применять методические разработки использования компьютерных технологий к условиям реального учебного процесса в образовательных учреждениях начального профессионального образования и т.д.

знать и уметь использовать:

– основные требования, содержание методики организации и проведения профессиональной подготовки рабочих;

– дидактические возможности, принципы действия, технологию использования и методику применения дидактических средств и т.д.

уметь:

– интерпретировать собственное психическое состояние, владеть простейшими приемами психической саморегуляции;

– направлять саморазвитие и самовоспитание личности и т.д.

– владеть:

– методиками проектирования, организацией проведения занятий по общетехническим и специальным предметам, практическому обучению в области информатики;

– методиками проектирования педагогических технологий и технико-методического обеспечения для подготовки современного рабочего и т.д.

Итоговая государственная аттестация педагога профессионального обучения включает выпускную квалификационную работу (дипломный проект или работу) и государственный экзамен по психолого-педагогической подготовке.

Дисциплина «Микропроцессорная техника» относится к дисциплинам отраслевой подготовки, но не входит в федеральный компонент, поэтому обязательный минимум содержания основной образовательной программы подготовки педагога профессионального обучения направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля подготовки «Информационные технологии» профилизации «Компьютерные технологии автоматизации и управления» разрабатывается непосредственно ВУЗом.

1.2 Учебный план

Учебный план – официальный документ, отражающий объем и содержание обучения.

Учебный план (типовой) – это документ рекомендательного характера, устанавливающий перечень и объем учебных циклов и предметов применительно к профессии и специальности с учетом ступени квалификации, минимального (базового) срока обучения и определяющий степень самостоятельности региональных органов образования, образовательных учреждений в разработке компонентов содержания обучения.

Учебный план (рабочий) – документ, разработанный для конкретного образовательного учреждения с учетом выбранной специальности и требований государственного, регионального и местного компонентов образования.

Перечень и блоки учебных предметов, их объем, последовательность изучения устанавливаются на основе ряда принципов, отражающих основные положения и закономерности формирования профессиональных знаний, умений, навыков. К ним относятся: научность, систематичность, последовательность, реализация межпредметных связей и т.д.

Согласно принципу систематичности все учебные планы имеют предметно-цикловую или блочно-модульную структуру, которая обеспечивает

возможность формирования системы научных знаний, умений и навыков, их логическую взаимосвязь, последовательность введения в учебный процесс.

Принцип последовательности в современных учебных планах подготовки специалистов выражается в ступенях профессионального обучения. Наиболее распространенными являются три ступени образования: начальное, среднее и высшее профессиональное образование.

Анализируя план, преподаватель или мастер определяет место предмета в системе изучаемых учащимися дисциплин и на этой основе планирует и реализует в учебном процессе межпредметные связи. Данные о количестве часов, выделенных на предмет, и их распределение на полугодия позволяют ему ориентировочно оценивать объем и сложность материала. О степени и значимости предмета можно судить по виду итогового контроля знаний учащихся (экзамен или зачет), перечню тех знаний и умений профессиональной характеристики, которые должны быть сформированы при изучении предмета [18].

Рассмотрев учебный план (ФГОС), составляем перечень предметов профессионально – технического цикла и определяем количество отведенных часов на изучение предметов в неделю, полугодие, год.

Структура учебных планов является основой для составления учебных программ по отдельным предметам, в которых находят свое отражение конкретные цели, задачи, принципы общеобразовательного и профессионального обучения, содержание предмета.

В учебном плане дисциплина «Микропроцессорная техника» входит в дисциплины специализации. Количество учебных часов аудиторных занятий (120 часов), лекций (40 часов), практических занятий (40 часов) и лабораторных работ (40 часов) определено кафедрой «Информационных систем и технологий».

1.3 Рабочая программа

Рабочая программа – нормативно-управленческий документ образовательного учреждения, характеризующий систему организации образовательной деятельности. Рабочая программа конкретизирует соответствующий образовательный стандарт с учетом необходимых требований к ее построению, а также описывает национально-региональный уровень, учитывает возможности методического, информационного, технического обеспечения учебного процесса, уровень подготовки учащихся, отражает специфику обучения в данном образовательном учреждении.

Таким образом, **рабочая программа** – это учебная программа, разработанная педагогом на основе Примерной для конкретного образовательного учреждения и определенного класса (группы), имеющая изменения и дополнения в содержании, последовательности изучения тем, количестве часов, использовании организационных форм обучения и т.п. [18].

Проанализировав пояснительную записку к рабочей программе по дисциплине «Микропроцессорная техника» определили цели изучения предмета:

- дать студентам наиболее общее представление об архитектуре всех основных микропроцессорных компонентов;
- познакомить студентов с принципами построения современных микропроцессорных систем;
- рассмотреть вопросы взаимодействия различных компонентов в микропроцессорных системах;
- изучить используемые на практике методики расчета микропроцессорных систем;
- сформулировать наиболее существенные требования и рекомендации по эксплуатации и техническому обслуживанию микропроцессорных систем.

Основные задачи дисциплины:

- в процессе обучения студенты должны изучить типовые структуры микропроцессоров, запоминающих устройств и внешних устройств ввода и вывода информации микропроцессорных систем, временные диаграммы их работы и алгоритмы функционирования;
- овладеть методикой построения различных узлов микропроцессорной системы с требуемыми параметрами и техническими характеристиками;
- научиться распределять, делить и расширять адресное пространство микро-ЭВМ, правильно осуществлять передачу информации в микропроцессорной системе, технически грамотно выполнять расчеты электрических нагрузок различных узлов и оценивать нагрузочную способность и надежность схемных конфигураций;
- подробно изучить технические характеристики и возможности использования современных интерфейсных БИС, нашедших наиболее широкое практическое применение в микропроцессорных системах.

В результате изучения дисциплины студенты должны уметь, исходя из особенностей решаемой задачи, сделать обоснованный выбор общей структуры микропроцессорной системы, организовать наиболее эффективные процедуры информационного обмена, правильно согласовать различные архитектурные характеристики и параметры взаимодействующих компонентов, предложить конкретный способ реализации отдельных узлов или системы в целом.

Для изучения дисциплины «Микропроцессорная техника» необходимы знания по следующим предметам:

- высшая математика (дифференциальное исчисление, неопределенный и определенный интегралы);
- физика (электростатика, постоянный электрический ток, электромагнитные колебания и волны, элементы физики твердого тела);

- электрические цепи и сигналы (линейные цепи постоянного и переменного тока, переходные процессы);
- арифметические и логические основы вычислительной техники (логические элементы, типовые функциональные схемы и узлы);
- электроника и микросхемотехника (полупроводниковые диоды и транзисторы, цифровые интегральные микросхемы, усилители электрических сигналов).

Тематический план изучения дисциплины «Микропроцессорная техника» содержит четыре основных темы. Проанализируем одну из тем, например: «Основные микропроцессорные компоненты».

Тема включает три подтемы:

- микропроцессоры;
- запоминающие устройства;
- внешние устройства ввода и вывода информации.

При изучении подтемы «Микропроцессоры», студент получает общие сведения о микропроцессорах их классификации и технической характеристики; методику выбора микропроцессора при построении микропроцессорной системы.

При изучении подтемы «Запоминающие устройства», студент получает общие сведения о запоминающих устройствах их классификации и технической характеристики; умения построения модуля памяти микропроцессорной системы.

При изучении подтемы «Внешние устройства ввода и вывода информации», студент получает общие сведения о внешних устройствах их классификации и технической характеристики; знания об интерфейсе сопряжения внешних устройств с шинами микропроцессорной системы.

Для преподавателя по данной теме рекомендуется учебное пособие для вузов: Пухальский Г. И. Проектирование микропроцессорных устройств и в

качестве методического пособия: Шалунова М. Г., Эрганова Н. Е. Практикум по методике профессионального обучения.

Общий объем учебной работы (в часах) составляет: аудиторных занятий 120, лекций 40, лабораторных 40. На самостоятельную работу студентам в рабочей программе 60 часов.

Так же на проведение лабораторных работ для студентов заочного обучения выделено небольшое количество часов.

1.4 Методические указания для выполнения лабораторных работ

Методические указания содержат описание учебного микропроцессорного комплекса (УМП), его состав и органы управления, режимы работы и директивы системной программы «МОНИТОР» УМК, а также в них представлена система команд микропроцессора КР580ВМ80

Методические указания включают пять лабораторных работ.

Лабораторная работа 1 «Изучение состава, назначения режимов работы УМП».

Цель: изучение назначения, конструкции и состава учебного микропроцессорного комплекса, режимов его работы и основных директив программы «МОНИТОР».

Лабораторная работа 2 «Изучение временных диаграмм выполнения команд загрузки и перемещения данных».

Цель: изучение особенностей функционирования микропроцессора при выполнении команд загрузки и перемещения данных, овладение методикой исследования временных диаграмм работы микропроцессора.

Лабораторная работа 3 «Изучение временных диаграмм выполнения команд ввода и вывода информации».

Цель: изучение особенностей функционирования микропроцессора при выполнении команд ввода и вывода информации, приобретение навыков программного управления клавиатурой и дисплеем УМК.

Лабораторная работа 4 «Изучение временных диаграмм выполнения команд арифметических и логических операций».

Цель: изучение особенностей функционирования микропроцессора при выполнении команд арифметических и логических операций, приобретение навыков реализации простейших вычислительных процедур.

Лабораторная работа 5 «Изучение временных диаграмм выполнения команд управления программой».

Цель: изучение особенностей функционирования микропроцессора при выполнении команд передачи управления, организация условных и безусловных переходов в программах, приобретение навыков модульного программирования.

Все лабораторные работы содержат общие сведения и порядок выполнения работы, включают исследование временных диаграмм выполнения команд и программирование вычислительных и управляющих процедур.

Данные методические указания предполагают самостоятельную подготовку каждого студента перед проведением лабораторных работ.

Методические указания на данный момент устарели и не несут необходимой современной информации, так как выпущены в 2012 году.

2 РАЗРАБОТКА ЛАБОРАТОРНЫХ РАБОТ

2.1 Понятие лабораторной работы

В системе работы по восприятию и усвоению нового материала учащимися широкое применение находит метод лабораторных работ. Свое название он получил от лат. *laborare*, что значит работать. На большую роль лабораторных работ в познании указывали многие выдающиеся ученые, например, Д.И. Менделеев отмечал, что в преддверии науки красуется надпись: наблюдение, предположение, опыт, указывая тем самым на важное значение опытных (лабораторных) методов познания.

В чем заключается сущность лабораторных работ как метода обучения? Лабораторная работа – это такой метод обучения, при котором студенты под руководством педагога и по заранее намеченному плану проделывают опыты или выполняют определенные практические задания. И в процессе их выполнения воспринимают и осмысливают новый учебный материал [15].

Проведение лабораторных работ, с целью осмысления нового учебного материала, включает в себя следующие методические приемы:

- постановку темы занятий и определение цели и задач лабораторных работ;
- краткие теоретические сведения;
- определение порядка лабораторных работ и отдельных ее этапов;
- непосредственное выполнение лабораторных работ студентами и контроль педагога за ходом занятий и соблюдением техники безопасности;
- подведение итогов лабораторных работ и формулирование основных выводов.

Лабораторные работы, как метод обучения, во многом носят исследовательский характер и в этом смысле высоко оцениваются в дидактике. Они

пробуждают у студентов глубокий интерес к окружающей природе, стремление осмыслить, изучить окружающие явления, применять добытые знания к решению и практических, и теоретических проблем. Метод этот воспитывает добросовестность в выводах, трезвость мысли. Лабораторные работы способствуют ознакомлению студентов с научными основами современного производства, выработке навыков обращения с реактивами, приборами и инструментами, создавая предпосылки для технического обучения.

В этом большую роль играют лабораторные практикумы и учебные эксперименты.

Учебный эксперимент является одним из важнейших методов обучения, источником знаний и средством наглядности одновременно. Он может использоваться в качестве введения к той или иной теме курса (мотивация), как иллюстрация к объяснению нового материала (восприятие и осмысление), как повторение или обобщение пройденного (интериоризация) или как контроль приобретенных знаний, умений, навыков, т.е. на всех этапах процесса обучения.

Различают следующие виды учебного эксперимента:

Демонстрационный эксперимент. Проводится в ходе объяснения для иллюстрации законов, явлений и их применения, демонстрации принципов действия технических установок, а также для знакомства учащихся с фундаментальными опытами. Готовится и выполняется только преподавателем.

Фронтальные лабораторные работы, опыты и наблюдения. (Выполняются на уроке всеми учащимися одновременно на однотипном оборудовании, под непосредственным руководством преподавателя).

Практикум. Является формой самостоятельной работы учащихся, готовится учащимися заранее и выполняется по письменной инструкции.

Внеурочные опыты и наблюдения. Разновидность домашнего задания. Используется для актуализации теоретических знаний или как мотивация к изучению нового материала.

Методически грамотно организованный эксперимент способствует как формированию практических умений, так и активизации теоретических знаний, полученных ранее. В процесс обучения вовлекаются различные каналы восприятия (слух, зрение, осязание, обоняние и т.д.). Это позволяет организовать полученную информацию, как систему ярких образов, и заложить ее в долговременную память.

С другой стороны, подготовка и проведение лабораторных работ являются довольно непростым делом и требуют от учителя знания некоторых методических особенностей, в значительной степени зависит от наличия тех или иных приборов и инструментов.

Частично эти и другие проблемы можно решить, используя при обучении компьютерный эксперимент. Под данным термином мы понимаем лабораторную работу, проводимую полностью на компьютере, без использования других технических средств обучения.

Некоторые эксперименты, проводимые в области компьютеризации учебного процесса, показывают, что применение компьютерного эксперимента позволяет существенно сократить время, которое тратится на рутинную работу (варьирование параметров эксперимента путем изменения схемы установки, расчета результатов измерения и т.д.), тем самым, высвобождая время для более серьезного уяснения целей и задач проводимого эксперимента.

Кроме того, появляется возможность продемонстрировать опыты, которые невозможно провести в условиях учебного кабинета.

Отметим характерные особенности компьютерного эксперимента.

Формой работы является диалог учащегося с компьютером. При этом в функции компьютера входит:

Реализация программными средствами модели изучаемого объекта, установки, процесса или ситуации

Имитация средств измерения и выполнение рутинной части обработки измерений.

Оценка действий учащегося.

Функции учащегося (несколько отличаются от его функций в традиционном эксперименте):

Анализ информации, которую программа выдает на экран дисплея.

Выбор условий эксперимента.

Проведение серий экспериментов для достижения цели, сформулированной в начале работы.

Корректировка последующих шагов с целью получения более высокой оценки и решения задачи более рациональным способом.

Разумеется, что возможна реализация эксперимента только в его компьютерном варианте.

В настоящее время имеются разработки виртуальных лабораторных практикумов, однако окончательно решение проблемы требует пристального внимания специалистов различных профилей, в том числе и психолого-педагогического.

Таким образом, целью проведения лабораторных работ является формирование умений студентов практической работы с микропроцессором, закрепление и конкретизацию полученных теоретических знаний, полное осуществление связи теории и практики в обучении.

2.2 Методика проведения лабораторных работ

Основная функция, которую выполняют лабораторные работы, – овладение студентами практических умений работы с вычислительной техники.

Анализируя содержание лабораторных работ по основам микропроцессорной техники, нетрудно заметить, что в качестве обобщенных умений выступают методы. Для того чтобы выполнить любую лабораторную работу по

основам микропроцессорной техники, необходимо изучить особенности функционирования микропроцессора, осуществить расчет и анализ режимов работы.

Планирование лабораторных работ осуществляется с помощью методических рекомендаций к проведению лабораторных работ по основам микропроцессорной техники. Рассмотрим традиционную структуру методики проведения лабораторных работ.

В методических рекомендациях даются:

- тема лабораторной работы из программы по предмету «Микропроцессорная техника»;
- цель лабораторной работы (нужно учесть, что формулировки целей часто расплывчаты и не нацеливают учащихся на конкретную деятельность);
- краткие теоретические положения (в этой части руководство к проведению лабораторной работы дублирует содержание учебника);
- перечень оборудования и аппаратуры для проведения лабораторной работы;
- принципиальная (монтажная) схема проведения лабораторного исследования;
- порядок выполнения, краткое описание приемов деятельности учащихся, формы представления результатов измерений (таблицы, диаграммы, графики);
- выводы по работе;
- контрольные вопросы.

Приведенная структура методических рекомендаций в настоящее время используется во всех типах учебных заведений (училищах, техникумах, вузах). В то же время, надо признать, она организует деятельность студентов, но не раскрывает логической последовательности выполнения операций и приемов проведения лабораторного исследования. Все ли студенты знакомятся с теоретическими положениями? Чаще всего все. Но студенты, при-

ступая к выполнению работы, не осознают того, что должны получить в результате исследования.

Чтобы изменить существующее положение, необходимо в руководстве к лабораторным работам раскрыть программу проведения исследования и всю последовательность выполняемых операций по определению параметров микропроцессорной системы.

Формируется цель проведения лабораторной работы. Затем указывается предмет исследования. Это необходимо для того, чтобы конкретизировать область исследования. Например, изучение структуры и режимов работы микропроцессора, учебной микро-ЭВМ и т.д. Далее определяется метод исследования.

Ведущими методами исследования в вычислительной технике являются: измерение, моделирование, анализ параметров, векторные диаграммы и т.д. Выбрав методы, студенты приступают к их выполнению.

После этого следует показать принципиальную схему исследования. Далее следует этап проведения лабораторной работы, который включает:

- краткие теоретические сведения;
- задание для выполнения;
- контрольные вопросы.

Следующий, основной этап – проведение лабораторной работы. Он включает в себя перечень заданий. Содержанию задания соответствуют методы исследования и основные элементы контроля. Этот этап лучше всего представить в виде технологической карты.

Заключительный этап лабораторной работы – подведение итогов лабораторных работ и формулирование основных выводов.

Изложенная методика проведения лабораторной работы имеет преимущество, которое заключается в том, что студенты выступают в роли исследователей. Содержание лабораторной работы включает систему умственных и практических действий по овладению методами исследования.

2.3 Разработка тем лабораторных работ

В настоящее время, студенты выполняют лабораторные работы на учебном микропроцессорном комплекте (УМПК-80) на базе микропроцессора КР580ВМ80А (Intel8080). Лабораторная установка состоит из учебного микропроцессорного комплекта (УМК), набора модулей, подключаемых к ее системной шине и различных периферийных устройств. УМК представляет собой учебную микро-ЭВМ, предназначенную для изучения программирования, проектирования и настройки микропроцессорных устройств и систем, выполненных на МП КР580ВМ80

Развитие информационных технологий привело к появлению понятия «виртуальный лабораторный практикум» (ВЛП), в основе которого лежит имитационное компьютерное моделирование. Основные способы использования ВЛП в учебном процессе:

- в качестве компьютерного «тренажера» для подготовки к выполнению практикума в реальной лаборатории (при этом программы компьютерного и физического экспериментов, как правило, одинаковы);
- как дополнение к реальному практикуму, предусматривающее такие компьютерные эксперименты, которые по различным причинам (техническим, финансовым, организационным и т.п.) не могут быть реализованы на физическом оборудовании.

Использование ВЛП в качестве компьютерного «тренажера» позволяет обучающемуся лучше подготовиться к проведению физического эксперимента, глубже уяснить исследуемые эффекты, приобрести навыки работы с измерительными приборами (в случае, если виртуальный практикум включает компьютерные модели измерительных приборов, близкие по своим свойствам к свойствам реальных приборов). Обычно такой подход можно рекомендовать для студентов заочно-дистанционной формы обучения, поскольку он не только способствует лучшему усвоению изучаемого материала, но и

позволяет сократить продолжительность выполнения практикума в реальной лаборатории в период пребывания в стенах учебного заведения.

Если ВЛП используется как дополнение к реальному практикуму, то он должен быть ориентирован на проведение исследований повышенного уровня сложности или исследований, требующих дорогостоящего оборудования, которым не располагает университет.

По технологиям создания ВЛП можно выделить следующие основные варианты:

- ВЛП на основе универсальных пакетов программ, обеспечивающих возможность применения в широком спектре предметных областей. Примером может служить система LabVIEW фирмы National Instruments. Универсальные пакеты содержат обширные библиотеки элементов, предназначенных для разработки виртуальных интерфейсов физических приборов и лабораторных установок;

- ВЛП на основе специализированных предметно-ориентированных пакетов программ, предназначенных для сравнительно ограниченного набора предметных областей. В качестве примера отметим систему Multisim фирмы Electronics Workbench, созданную для моделирования электронных схем, систему ChemOffice фирмы CambridgeSoft, предназначенную для моделирования и анализа химических процессов и т.п. Также как и в предыдущем случае, программное обеспечение данного класса представляет собой универсальную среду, предназначенную для решения прикладных задач пользователя;

- ВЛП на основе Java-апплетов. В отличие от предыдущих случаев, где пользователь (преподаватель) обычно работает в режиме графического программирования, процесс создания Java-апплетов является гораздо более трудоемким и требует программирования в кодах. Тем не менее, данная технология имеет и определенные достоинства, особенно когда речь заходит о ВЛП, предназначенном для сетевого применения. Так, например, приложе-

ния, создаваемые в системе LabVIEW, занимают примерно 2,5-3 Мбайт памяти, а типичный размер виртуальной лабораторной работы на основе Java-апплета – десятки-сотни килобайт.

Развитие сетевых компьютерных технологий привело к появлению лабораторного практикума, реализуемого в режиме удаленного доступа к реальному оборудованию. Учитывая, что реализация удаленного доступа к реальному оборудованию связана с решением ряда проблем (необходимостью сопряжения лабораторного макета с ПК, обеспечением надежной защиты оборудования от возникновения аварийных режимов, низкой эффективностью использования оборудования из-за невозможности в ряде случаев реализовать коллективный доступ и т.д.), данная технология имеет достаточно много оппонентов. Тем не менее, она тоже имеет свое право на существование, а в ряде случаев имеет очевидные преимущества перед ВЛП.

Важным является вопрос, не является ли ВЛП альтернативой реальному лабораторному практикуму. С одной стороны, современные компьютерные технологии имитационного моделирования позволяют создавать виртуальные интерфейсы реального лабораторного оборудования, воспроизводящие и внешний вид, и его параметры с очень высокой точностью. С другой стороны, поддержание в рабочем состоянии и своевременное обновление лабораторного оборудования, включая и измерительные приборы, требует немалых финансовых средств.

Тем не менее, любой, даже сколь угодно высококлассный ВЛП, в большинстве случаев не заменит по своему обучающему воздействию, оказываемому на студента, работу с реальным оборудованием.

Одним из таких ВЛП является программная модель стенда УМПК-80, для которой и были разработаны темы лабораторных работ.

2.4 Лабораторный практикум

2.4.1 Лабораторная работа № 1 «Изучение интерфейса и режима работы программной модели стенда УМПК-80»

Цель работы: изучение интерфейса программной модели стенда УМПК-80

Оборудование: персональный компьютер, программная модель стенда УМПК-80.

Интерфейс программной модели стенда УМПК-80

Внешний интерфейс пользователя программной модели стенда УМПК-80

Программная модель стенда УМПК-80 представляет собой средство, которое дает гораздо более широкие и удобные возможности для набора и отладки программ (к примеру, может быть возможен одновременный просмотр всех регистров, памяти, ввод команд в мнемонических обозначениях и в шестнадцатеричных кодах, ассемблирование и дизассемблирование команд и т. д.).

Знакомство с программой следует начать с изучения рабочего окна. Для этого откройте в вашем компьютере файл с программной моделью стенда УМПК-80. На экране монитора появится диалоговое окно программы (рисунк 1).

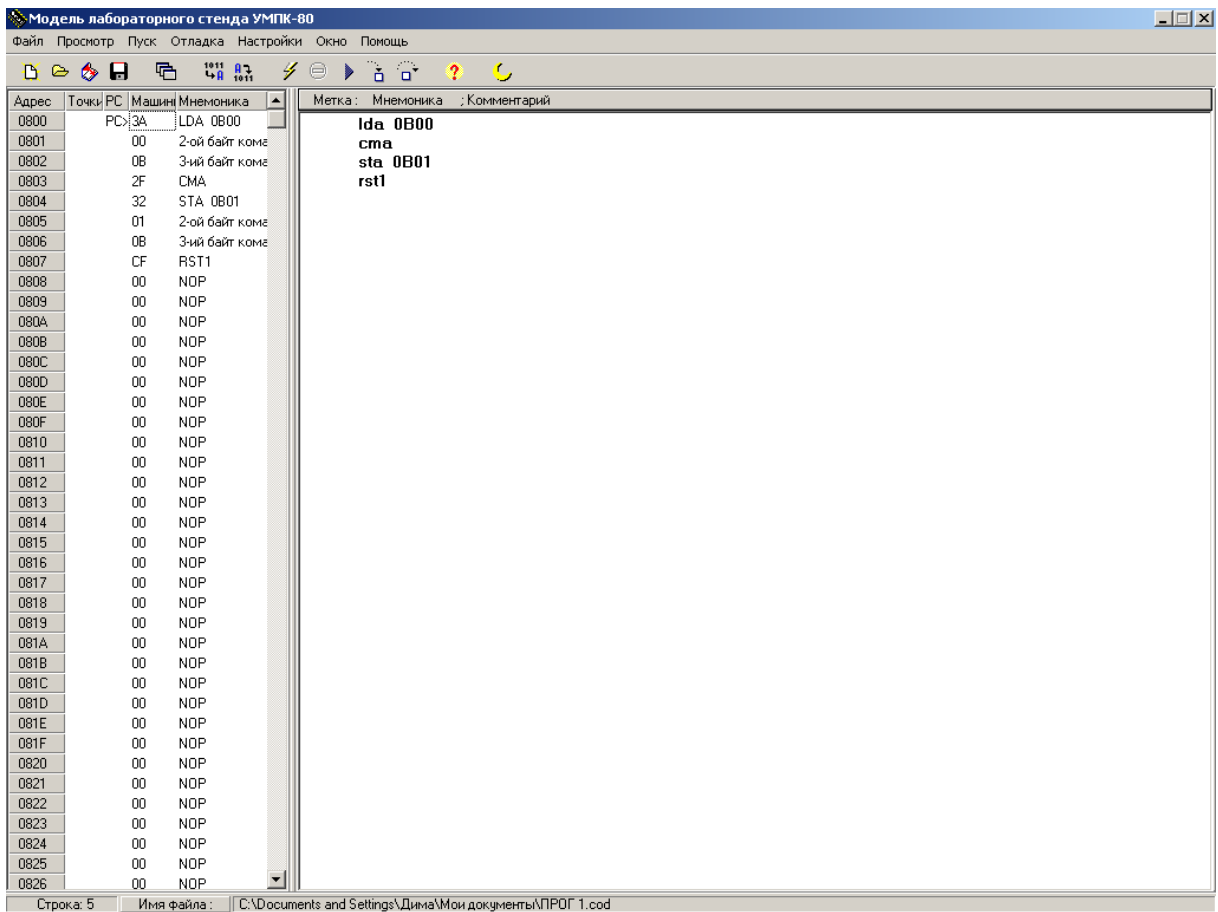


Рисунок 1 – Внешний вид экрана компьютера при работе с программной моделью стенда УМПК-80

МЕНЮ МОДЕЛИ

Это меню находится в верхней части главного окна модели. Оно содержит все команды необходимые для работы с модель (рисунок 2).



Рисунок 2 – Внешний вид меню моделей.

В него входят:

- подменю «Файл», содержит команды для работы с файлами;

Подменю «Файл» находится в главном меню модели и вызывается сочетанием клавиш «Alt+Ф». В нем находятся команды для работы с файлами (таблица 1).

Таблица 1 – Таблица команд подменю «Файл»

Команда	Описание действий при выборе команды
Новый	Закрывается текущая ИПП. Если в ней были не сохраненные данные, то предлагается записать их. Создается пустая ИПП.
Открыть	Закрывается текущая ИПП. Если в ней были не сохраненные данные, то предлагается записать их. Открывается файл с ИПП, ранее сохраненный на диске.
Открыть вновь	Модель может запоминать до 10 имен последних открытых файлов. В этом случае по команде «Открыть вновь» открывается их список, из которого можно выбрать нужное. После чего происходит следующее: закрывается текущая ИПП, и если в ней были не сохраненные данные, то предлагается записать их; открывается выбранный файл, если он найден.
Сохранить	Текущая ИПП записывается в файл на диск. Если ИПП ранее не записывалась, то предлагается ввести имя файла.
Сохранить как *.asm	Сохраняются мнемонические коды ИПП в ASM-файле.
Сохранить как *.cod	Сохраняют машинные коды ИПП в COD-файле.
Закрыть	Закрывается текущая ИПП. Если в ней были не сохраненные данные, то предлагается записать их.
Выход	Закрывается текущая ИПП. Если в ней были не сохраненные данные, то предлагается записать их. Модель завершает свою работу.

– подменю «Просмотр», содержит команды для управления выводом информации;

Подменю «Просмотр» находится в главном меню модели и вызывается сочетанием клавиш «Alt+П». В нем находятся команды для работы с визуальными объектами модели (окна, панель и т. п.) (таблица 2).

Таблица 2 – Таблица команд подменю «Просмотр»

Команда	Описание действий при выборе команды
Регистры и флаги	Открывается и активизируется окно «Регистры и флаги».
ОЗУ стенда	Открывается и активизируется окно «ОЗУ стенда».
Стек	Открывается и активизируется окно «Стек».
Порты	Открывается и активизируется окно «Порты».
Средства стенда	Открывается и активизируется окно «Средства стенда».
Все	Открываются и активизируются окна: «Регистры и флаги», «ОЗУ стенда», «Стек», «Порты», «Средства стенда».
Панель инструментов	Выводится (скрывается) панель инструментов. При галочке справа от команды панель инструментов видна.
Дешифрация кодов	Если справа от команды стоит галочка, то в секции машинных кодов производится дизассемблирование hex-кодов, начиная с адреса в счетчике команд.
Просмотр сообщений	Если справа от команды стоит галочка, то видна секция сообщений.

– подменю «Пуск», содержит команды для выполнения и обработки исследуемой программы;

Подменю «Пуск» находится в главном меню модели и вызывается сочетанием клавиш «Alt+У». В нем находятся команды, необходимые для запуска ИПП в автоматическом режиме, а также ее перевода в различные форматы кодов (hex-коды или мнемокоды) (таблица 3).

Таблица 3 – Таблица команд подменю «Запуск»

Команда	Описание действий при выборе команды
Выполнить	ИПП запускается на выполнение в автоматическом режиме, с адреса, указанного в РС. Необходимо помнить, что ПРОГРАММА ВСЕГДА ВЫПОЛНЯЕТСЯ ПО МАШИНЫМ КОДАМ. Поэтому для выполнения программы в мнемосокодах необходимо предварительно произвести ее ассемблирование.
Ассемблирование	ИПП переводится из мнемосокодов в машинные коды. Мнемосокоды берутся из секции мнемосокодов, а переведенные hex-коды записываются в секции машинных кодов, с адреса указанного в настройках модели (см. раздел «Настройки»).
Дизассемблирование	ИПП перевод из машинных кодов в мнемосокоды. Hex-коды берутся из секции машинных кодов, начиная с адреса указанного в настройках модели (см. раздел «Настройки»), и переведенные мнемосокоды записываются в секции мнемосокодов.
Сброс процессора	Производится сброс регистров процессора. В РОН В,С,D,E,H,L и аккумулятор А заносится значение 00h; РС устанавливается на адрес 0800h, а SP на адрес 0B00h.

– подменю «Отладка», содержит команды для отладки исследуемой программы;

Подменю «Отладка» находится в главном меню модели и вызывается сочетанием клавиш «Alt+O». В нем находятся команды, необходимые для отладки ИПП в пошаговых режимах (таблица 4)

Таблица 4 – Таблица команд подменю «Отладка»

Команда	Описание действий при выборе команды
Шаг команды	ИПП запускается на выполнение с адреса, указанного в РС. Программа выполняется по командам, т.е. после выполнения одной команды происходит останов ИПП.
Машинный цикл	ИПП запускается на выполнение с адреса, указанного в РС. Программа выполняется по машинным циклам, т.е. после каждого цикла выполнение ИПП приостанавливается. Выполнение программы по машинным циклам сопровождается выводом отладочной информации в окно «средства стенда»
Точка останова	Устанавливается (удаляется, если уже стоит) точка останова, т.е. при выполнении ИПП в автоматическом режиме, в этой точке произойдет останов программы. Адрес точки останова определяется курсором (пунктирная рамка) в секции машинных кодов. Установка точки также возможна непосредственно в самой секции машинных кодов. Для этого необходимо щелкнуть левой кнопкой мыши в нужной строке таблицы и столбце «точки останова».
Удалить все точки	Удаляются все точки останова.
Останов программы	Прекращается выполнение ИПП в автоматическом режиме.

– пункт «Настройки», открывает окно настроек модели;

Команда «Настройки» находится в главном меню модели и вызывается сочетанием клавиш «Alt+N». По этой команде вызывается окно настроек модели.

В нижней части окна расположены три кнопки (таблица 5)

Таблица 5 – Таблица кнопок пункта «Настройки»

Кнопка	Описание действия при нажатии кнопки
По умолчанию	Устанавливает во всех параметрах модели значения по умолчанию. Для их записи необходимо нажать кнопку «ОК».
ОК	Закрывает окно «настройки», при этом изменения в параметрах модели не сохраняются.
Отмена	Закрывает окно «настройки», при этом изменения в параметрах модели сохраняются.

В верхней части окна находятся пять страниц, на которых находятся поля для изменения настроек модели.

Страница «компиляция»

На этой странице устанавливается адрес, начиная с которого будет производиться дизассемблирование программы, и начиная с которого будут записываться машинные коды проассемблированной программы.

Страница «программа»

Здесь устанавливается приоритет ИПП при ее выполнении в автоматическом режиме. Для автоматического режима ИПП запускается как отдельный процесс. При большой загрузке компьютера задачами высоким приоритет этого процесса может повлиять на скорость его выполнения.

Страница «стенд»

Здесь выбирается моделируемый стенд. Возможно выбрать моделирование «нового», либо «старого» стенда. Их отличие состоит в различных адресах портов ввода/вывода.

Страница «RST-адреса»

В стенде УМПК-80 по команде RST<N> производится вызов прерывающей программы в ПЗУ, по адресу то 0000h до 0038h, в зависимости от N. В данной модели ПЗУ не моделируется, поэтому, чтобы хоть как-то использовать RST-команды здесь можно разместить прерывающую программу в ОЗУ. Для этого в окне «настройки», на странице RST-адреса необходимо установить ее адрес. Если прерывающей программы нет, то необходимо вместо адреса вызова ввести RET, для этого в соответствующем поле адреса нужно задать значение 0BB0h.

Страница «автоматизация»

На этой странице можно включить режимы автоматического ассемблирования (дизассемблирования) открываемого файла, в зависимости от его расширения. При расширении «COD» производится дизассемблирование файла, в противном случае – ассемблирование. О включении того или иного режима сигнализирует галочка в соответствующем поле, установка которой производится щелчком левой кнопки мыши на нужном поле.

– подменю «Окно», содержит команды для управления окнами модели. Подменю «Окно» находится в главном меню модели и вызывается сочетанием клавиш «Alt+K». В нем находятся команды, управления окнами (таблица 6).

Таблица 6 – Таблица команд подменю «Окно»

Команда	Описание действий при выборе команды
Закреть все	Закрываются окна: «Регистры и флаги», «ОЗУ стенда», «Стек», «Порты», «Средства стенда». Для закрытия текущего окна используйте сочетание клавиш «Alt+F4».

– подменю «Помощь», содержит команды для вызова справочной информации.

Подменю «Помощь» находится в главном меню модели и вызывается сочетанием клавиш «Alt+M». В нем находятся команды управляющие выводом справочной информации (таблица 7).

Таблица 7 – Таблица команд подменю «Помощь»

Команда	Описание действий при выборе команды
Инструкция пользователя	Вызывает все разделы справочной системы.
Описание стенда УМПК-80	Вызывает разделы справочной системы, описывающие лабораторный стенд УМПК-80.
Система команд МПК580ВМ80	Вызывает разделы справочной системы, описывающие систему команд микропроцессора К580ВМ80.
Сообщения об ошибках	Вызывает разделы справочной системы, описывающие сообщения об ошибках, возникающих при исследовании программы пользователя.
О программе	Вызывает окно с кратким описанием назначения модели.

ПАНЕЛЬ ИНСТРУМЕНТОВ

Панель инструментов (рис 1.3) предназначена для быстрого доступа к наиболее часто используемым командам в меню и содержит:

- создание нового файла;
- открытие файла;
- закрытие файла;
- сохранение файла на диск;
- открытие всех окон;

- дизассемблирование исследуемой программы;
- ассемблирование исследуемой программы;
- сброс процессора;
- останов выполнения исследуемой программы;
- выполнение исследуемой программы в автоматическом режиме
- выполнение исследуемой программы по командным циклам;
- выполнение исследуемой программы по машинным циклам;
- вызов справочной информации;
- выход из модели.

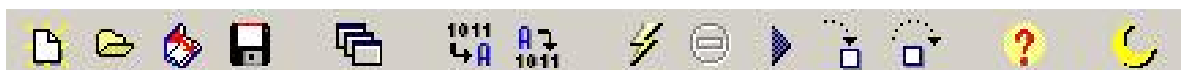


Рисунок 3 – Панель инструментов

Панель инструментов можно отключить (подключить), выбрав в главном меню программы пункт «Просмотр / Панель инструментов». При галочке справа от команды панель инструментов видна.

СЕКЦИЯ МАШИННЫХ КОДОВ

Общие сведения о секции машинных кодов

Эта секция главного окна модели предназначена для ввода и редактирования ИПП в машинных кодах, и представляет собой таблицу, каждая строка которой соответствует одной ячейке ОЗУ. Столбцы таблицы имеют следующее назначение (таблица 8)

Для занесения значения в ячейку нужно клавишами управления или мышью установить на нее курсор (пунктирная рамка) и набрать новое значение, после чего подтвердить ввод клавишей «enter». Необходимо заметить, что всегда нужно заносить оба разряда числа, т.е. для ввода значения, например, «7» или «F» нужно соответственно набрать «07», «0F».

Таблица 8 – Таблица секции машинных кодов

Заголовок столбца	Назначение столбца
Адрес	Содержит адреса ячеек памяти.
Точки останова	Отображает установленные точки останова
РС	Отображает счетчик команд в виде «РС»
Машинный код	Содержит значение ячейки памяти по адресу указанному в столбце «адрес».
Мнемоника	Содержит мнемоническое описание значения ячейки.

В секции машинных кодов присутствует локальное меню – меню секции машинных кодов, которое вызывается щелчком правой кнопки мыши. Оно содержит дополнительные команды редактирования программы в hex-кодах.

Для отладки программы удобно использовать точки останова, т.е. точки, в которых происходит остановка выполнения программы. Для ее установки нужно щелкнуть левой кнопкой мыши на пересечении столбца «точки останова» и строки с нужным адресом ячейки. Точку останова также можно установить, выбрав пункт «отладка | точка останова» в меню модели или нажать клавиши «Ctrl+F», при этом точка будет поставлена в текущей позиции курсора секции машинных кодов. Для удаления одной точки останова необходимо повторить вышеописанные действия. Для снятия всех точек используйте пункт меню «отладка | удалить все точки».

Меню секции машинных кодов

Активизация меню происходит по нажатию клавиши «F10» или щелчком левой кнопки мыши на соответствующем пункте. Также можно выбрать нужный пункт меню сочетанием клавиши «Alt» и клавиши соответствующей

подчеркнутой букве в названии пункта меню. Перемещение между пунктами осуществляется клавишами управления курсором.

Это меню вызывается щелчком правой кнопки мыши в секции машинных кодов. Оно содержит дополнительные команды редактирования программы в hex-кодах. Состав меню (таблица 9)

Таблица 9 – Таблица команд подменю

Команда	Описание действий при выборе команды
Вставить байт	Вставляет байт 00h в текущей позиции курсора секции машинных кодов. Все данные лежащие ниже этой точки смещаются вперед. При выполнении команды пересчет адресов перехода НЕ ПРОИЗВОДИТСЯ.
Удалить байт	Удаляет байт в текущей позиции курсора секции машинных кодов. Все данные лежащие ниже этой точки смещаются назад. При выполнении команды пересчет адресов перехода НЕ ПРОИЗВОДИТСЯ.
Удалить все	По всем адресам ОЗУ от 0800h до 0B00h заносится значение 00h, т.е. фактически удаляется программа, написанная в машинных кодах.

СЕКЦИЯ МНЕМОКОДОВ

Общие сведения о секции мнемкокодов

Эта секция главного окна модели предназначена для ввода и редактирования ИПП в мнемонических кодах, и представляет собой обычный текстовый редактор.

Перемещения внутри редактора осуществляются клавишами управления курсором.

Секция имеет локальное меню секции мнемкокодов, которое вызывается щелчком правой кнопки мыши. Оно содержит дополнительные команды редактирования программы в мнемонических кодах.

При вводе программы необходимо соблюдать следующий порядок: сначала пишется имя метки с двоеточием в конце, затем команда и далее комментарий. Допускается отсутствие в строке любой из этих частей.

Под меткой понимается расположенная в одной строке последовательность символов, заканчивающаяся двоеточием. В состав имени могут входить прописные и строчные буквы латинского алфавита, цифры и символ подчеркивания.

Комментарием считается текст от символа точки с запятой до конца строки.

Меню секции мнемкокодов

Это меню вызывается щелчком правой кнопки мыши в секции мнемкокодов. Оно содержит дополнительные команды редактирования программы в мнемкокодах. Состав меню (таблица 10)

Таблица 10 – Таблица команд подменю

Команда	Описание действий при выборе команды
Вырезать	Вырезается в буфер выделенный фрагмент текста в секции мнемкокодов.
Копировать	Вырезается в буфер выделенный фрагмент текста в секции мнемкокодов.
Вставить	Вырезается в буфер выделенный фрагмент текста в секции мнемкокодов.
Выделить все	Вырезается все содержимое секции мнемкокодов.
Удалить строку	Удаляется текущая строка в секции мнемкокодов. Текущей является та строка, в которой находится курсор.
Вставить команду	В текущей позиции курсора вставляется строка с командой. Команда выбирается из списка вида:
Удалить все	Удаляется все содержимое секции мнемкокодов.

СЕКЦИЯ СООБЩЕНИЙ

Общие сведения о секции сообщений

В эту секцию поступают сообщения об ошибках, возникающих в результате ассемблирования, дизассемблирования и выполнения программы. Перейти в нее можно с помощью клавиши «Tab» или щелчком левой кнопки мыши. Установив курсор на одно из сообщений можно перейти на строку с ошибкой, для этого нужно дважды щелкнуть левой кнопкой мыши на соответствующем сообщении или с помощью меню секции сообщений, вызвав его щелчком правой кнопкой мыши. Секцию сообщений можно убрать, выбрав пункт «закрыть» вышеуказанного меню или через меню модели (пункт «просмотр | просмотр сообщений»).

Меню секции сообщений

Это меню вызывается щелчком правой кнопки мыши в секции сообщений. Оно содержит команды необходимые для управления этой секцией. Состав меню (таблица 11)

Таблица 11 – Таблица команд меню секции сообщений

Команда	Описание действий при выборе команды
Перейти на строку с ошибкой	Осуществляется переход на строку, в которой возникла ошибка.
Очистить	Удаляется содержимое секции сообщений.
Закрыть	Закрывается секция сообщений.

Если при ассемблировании, дизассемблировании или выполнении программы пользователя встречаются недопустимые ситуации, то открывается

секция сообщений, в которую выводится сообщение об ошибке, вида: «ошибка в строке (по адресу) N: «описание _ ошибки»», где N -- номер строки или адрес ячейки ОЗУ, а «описание _ ошибки» – краткое пояснение ошибки (таблица 12).

Примечание: при возникновении ошибок в написании команд рекомендуется ознакомиться с разделом «таблица команд».

СТРОКА СОСТОЯНИЯ

Строка состояния находится в нижней части главного окна модели. Она состоит из двух частей. В левой ее части отображается текущая позиция курсора в секции мнемокодов, в правой – имя текущего файла.

Дополнительные окна

Регистры и флаги

Окно «регистры и флаги» вызывается командой «просмотр | регистры и флаги» в главном меню модели или сочетанием клавиш «Ctrl+R». В нем находятся регистры микропроцессора K580BM80. Окно разбито на три части – секции: «регистр признаков», «указатели», «регистры РОН» (таблица 13)

Перемещения по секциям и их элементам осуществляется клавишами «Tab» (вперед) и «Shift+Tab» (назад) или с помощью мыши. Если к полю ввода подвести и задержать на несколько секунд указатель мыши, то в появившейся подсказке будут находиться десятичное и двоичное представления значения этого поля.

Таблица 12 – Таблица сведений об ошибке

Сообщение об ошибке	Описание ошибки и причин ее возникновения
Мнемокод написан неправильно	Допущена синтаксическая ошибка в написании мнемокода.
Некорректный операнд	В написании мнемоники найден операнд не характерный для нее, например, для команды MOV, вместо регистра указана константа.
Лишний операнд	В написании мнемоники найден лишний операнд. Удалите его.
Отсутствует операнд	В написании мнемоники найден лишний операнд. Удалите его.
Отсутствуют операнды	В написании мнемоники не хватает одного операнда. Допишите его.
Команда ссылается на несуществующую метку	В написании мнемоники не хватает двух операндов. Допишите их через запятую.
Команда с таким машинным кодом отсутствует	В системе команд МП K580BM80 есть двенадцать шестнадцатеричных кодов, которые не используются.
Обращение за область ОЗУ при выполнении команды	При выполнении программы произошла попытка чтения или записи данных по адресу вне диапазона 0800h – 0BAF.
Не полный код команды	Возникает когда после записи в стек часть двухбайтной или трехбайтной команды попадает в область стека.
Переход в область стека	В данной модели запрещены переходы в область стека.
Невозможно изменить значение указателя стека	Возникает при попытке изменить значение указателя стека SP на адрес лежащий вне диапазона (PC+1) – 0BB0h.
Нет данных в стеке	Возникает при попытке выполнить команду <POP>, когда в стеке нет данных.
Невозможно изменить значение счетчика команд	Возникает при попытке изменить значение счетчика команд PC на адрес лежащий вне диапазона 0800h – (SP-1).
Переполнение стека	Переполнением стека в данной модели считается ситуация, когда при инкременте указателя стека SP полученный адрес равен адресу в счетчике команд PC.

Таблица 13 – Таблица команд окна «регистры и флаги»

Секция	Назначение секции
«Регистр признаков»	В ней находится регистр признаков МП, в котором содержатся пять флагов: S, Z, AC, P, C. С правой стороны в секции находится шестнадцатеричное значение регистра признаков, а с левой – его побитовое представление. Установка нового значения производится в поле ввода «FL».
«Указатели»	Здесь находятся указатель стека SP и счетчик команд PC. При вводе их новых значений необходимо иметь ввиду, что в поле ввода нельзя набрать более четырех символов, а также новое значение должно лежать в диапазоне адресов:– для счетчика команд [0800h – (SP-1)];– для указателя стека [(PC+1) – 0BB0h]. При нарушении этого условия восстановится значение, которое было до начала редактирования поля ввода.
«Регистры РОН»	В этой секции содержатся: аккумулятор (A), шесть регистров общего назначения (B, C, D, E, H, L), а также значение ячейки по адресу содержащемуся в регистровой паре HL, которая не является элементом микропроцессора и находится в секции только для удобства исследования команд МП использующих косвенную адресацию через регистровую пару HL.

ОЗУ стенда

Окно «ОЗУ стенда» вызывается командой «просмотр | ОЗУ стенда» в главном меню модели или сочетанием клавиш «Ctrl+M». В нем можно просмотреть любой фрагмент памяти. Содержимое памяти отображается в виде таблицы, в каждой строке которой показывается по шестнадцать байт, так адрес ячейки, лежащей на пересечении строки «0820h» и столбца «D», равен «082Dh». Основной функцией окна является просмотр памяти, однако, при желании можно отредактировать значение любой ячейки. Для этого необходимо клавишами управления курсором или мышью подвести курсор (пунктирная рамка) к ячейке памяти, значение которой Вы хотите исправить и нажать клавишу «enter» или дважды щелкнуть на ней левой кнопкой мыши.

При этом активизируется редактор ОЗУ, в котором нужно ввести новое значение ячейки, после чего редактор надо закрыть.

Стек

Окно «стек» вызывается командой «просмотр | стек» в главном меню модели или сочетанием клавиш «Ctrl+S». В нем можно просмотреть фрагмент памяти относящийся к стеку, т. е. ячейки с адресами от 0BB0h до адреса, на который указывает указатель стека SP. Содержимое памяти отображается в виде таблицы, в каждой строке которой показывается по одному байту. Таблица имеет три столбца. В первом столбце находятся адреса ячеек, в третьем – их значения. Во втором столбце находится визуальное отображение указателя стека. Основной функцией окна является просмотр памяти, однако, при желании можно отредактировать значение любой ячейки. Для этого необходимо клавишами управления курсором или мышью подвести курсор (пунктирная рамка) к ячейке памяти, значение которой Вы хотите исправить и нажать клавишу «enter» или дважды щелкнуть на ней левой кнопкой мыши. При этом активизируется редактор ОЗУ, в котором нужно ввести новое значение ячейки, после чего редактор надо закрыть.

Порты

Окно «порты» вызывается командой «просмотр | порты» в главном меню модели или сочетанием клавиш «Ctrl+P». В нем можно просмотреть содержимое портов ввода/вывода. В окне находятся две странички, на одной из которых находятся порты ввода, а на другой – порты вывода. Для перехода на нужную страничку нужно щелкнуть левой кнопкой мыши на закладке с соответствующим названием. Для редактирования значения определенного порта необходимо подвести курсор к ячейке с нужным адресом порта, ввести

новое значение и нажать клавишу «enter». Необходимо иметь в виду, что строка соответствует старшей части адреса порта, а столбец – младшей, т. е. адрес порта, лежащего на пересечении строки «2» и столбца «А», равен «2А».

Перемещения между портами осуществляется клавишами управления курсором.

Средства стенда

Окно «средства стенда» вызывается командой «просмотр | средства стенда» в главном меню модели или сочетанием клавиш «Ctrl+I». В нем находятся средства, относящиеся к лабораторному стенду, а именно (слева направо, сверху вниз) (таблица 14).

Таблица 14 – Таблица команд окна «средства стенда»

Элемент стенда	Описание элемента
Шина адреса	Имитирует светодиодную индикацию шины адреса стенда.
Шина управления	Имитирует светодиодную индикацию шины данных стенда.
Шина данных	Имитирует светодиодную индикацию шины управления стенда.
Дисплей	Имитирует шестиразрядный восьмисегментный дисплей стенда.
Порт ввода	Имитирует порт ввода с переключателей. Для переключения любого из них необходимо дважды щелкнуть на нем левой кнопкой мыши.
Клавиатура	Имитирует клавиатуру стенда.
Порт вывода	Имитирует порт вывода на светодиодные индикаторы.

Примечание:

И клавиатура, и дисплей могут использоваться только в программе пользователя. ВВОДИТЬ ПРОГРАММУ В МОДЕЛЬ С НИХ НЕЛЬЗЯ.

Редактор ОЗУ

Окно «редактор ОЗУ» вызывается из окон «ОЗУ стенда» и «Стек», так как их основная функция – просмотр памяти. Редактор предназначен для редактирования памяти. Редактирование производится по одному байту. Адрес текущей ячейки отображается слева от надписи «адрес ОЗУ». Редактор имеет четыре кнопки (таблица 15).

Таблица 15 – Таблица команд окна редактор ОЗУ

Кнопка	Описание действий при нажатии на кнопку
Зап.+Увел.	Производится запись отредактированного значения ячейки по текущему адресу, после чего текущий адрес увеличивается на единицу.
Вперед	Текущий адрес увеличивается на единицу без записи нового значения ячейки.
Назад	Текущий адрес уменьшается на единицу без записи нового значения ячейки.
Закреть	Закрывается редактор ОЗУ.

Для редактирования ячейки необходимо в поле ввода установить новое значение ячейки памяти, после чего подтвердить ввод нажатием кнопки «Зап.+Увел.».

Задание для домашней подготовки: по окончании лабораторной перейти к выполнению следующей лабораторной работы.

2.4.2 Лабораторная работа № 2 «Изучение выполнения команд загрузки и перемещения данных»

Цель работы: исследование выполнения отдельных команд МП и написание простых программ; освоение прямого и косвенного способов адресации в программах; формы записи программ.

Оборудование: персональный компьютер, программная модель стенда УМПК-80.

Краткие сведения из теории

Микропроцессор КР580ВМ80А имеет фиксированный набор команд. Время выполнения команды определяется процессом получения, декодирования и выполнения команды. В зависимости вида команды время ее выполнения может составлять 1 – 5 машинных циклов. Машинный цикл состоит из 3 – 5 машинных тактов.

На каждом машинном цикле МП проверяет состояние сигнала «Готов» на своем входе. При нулевом сигнале на этом входе работа МП приостанавливается. В программной модели УМПК 80 это используется для выполнения программы в пошаговом режиме. Одно нажатие кнопки «Шаг команды» приводит к выполнению одной команды. Одно нажатие кнопки «Машинный цикл» приводит к выполнению одного машинного цикла программы и вызова дополнительного окна «Средства стенда».

Рассмотрим, как записывается программа. Вначале программу записывают на языке Ассемблер, программа 1 (таблица 16).

Эта программа извлекает число из адреса памяти 0900, инвертирует его и записывает результат по адресу 0901.

Таблица 16 – Запись программы в мнемосокодах языка Ассемблер

Метка	Мнемосокод	Комментарий
Label1:	LDA 0900	Считать число из ячейки памяти с адресом 0900 в аккумулятор
	CMA	Инвертировать полученное число
	STA 0901	Записать результат из аккумулятора в ячейку памяти с адресом 0901
	JMP Label1	Безусловный переход на метку

Команды в программе данного МП могут быть одно-, двух- или трехбайтные и могут занимать соответственно одну, две или три ячейки памяти, программа 2 (таблица 17).

Таблица 17 – Размещение программы по адресам памяти

Адрес	Число	Комментарий
0800	3A	Код команды LDA
0801	00	Младший байт адреса
0802	09	Старший байт адреса
0803	2F	Код команды CMA
0804	32	Код команды STA
0805	01	Младший байт адреса
0806	09	Старший байт адреса
0807	C3	Код команды JMP
0808	00	Младший байт адреса
0809	00	Старший байт адреса

Запись программ удобно проводить в более компактной форме. В программе указывается начальный адрес каждой команды, но при этом понимается, что в зависимости от длины (одно-, двух- или трехбайтная) команды будут занимать от одной до трех последовательных ячеек. При такой записи в левом столбце указываются лишь начальные адреса команд в программе.

Это позволяет сократить объем при описании программ и сделать более простым их анализ, программа 3 (таблица 18).

Таблица 18 – Окончательный вид записи программы

Адрес	Код	Метка	Мнемокод	Комментарий
0800	3A 00 09		LDA 0900	Считать число из ячейки памяти с адресом 0900 в аккумулятор ($M_{0900} \rightarrow A$)
0803	2F		CMA	Инвертировать полученное число ($A = \bar{A}$)
0804	32 01 09		STA 0901	Записать результат из аккумулятора в ячейку памяти с адресом 0901 ($A \rightarrow M_{0901}$)
0807	C3 00 00		JMP 0000	Передача управления операционной системе «Монитор»

Программа 3, которая была рассмотрена выше, использует прямой способ адресации данных. При этом способе адрес ячейка памяти, с которой производятся определенные действия, указывается непосредственно в команде, которая эти действия производит. Например, в программе 3 это команды пересылок:

- LDA 0900 – считывание данных из памяти;
- STA 0901 – запись данных в память.

Кроме прямого способа адресации существует косвенный способ адресации. При этом способе адрес ячейки памяти, с которой предполагается произвести какие-либо действия, предварительно помещается в одну из 3-х регистровых пар HL, DE, BC. Наиболее удобно для этой цели использовать регистровую пару HL, т.к. есть ряд команд, которые по умолчанию работают с ней.

Далее, при необходимости обратиться к ячейке памяти, адрес которой записан в регистровой паре, используют соответствующие команды. Косвенный способ адресации позволяет экономить место в программе, т.к. команды в этом случае становятся из 3-х байтовых – однобайтовыми.

Пример использования косвенного способа адресации приведен в программе 4 (таблица 19). Эта программа совершает те же действия, что и программа 1.

Таблица 19 – Иллюстрация косвенного способа адресации

Адрес	Код	Метка	Мнемокод	Комментарий
0800	21 00 09	Label1:	LXI H, 0900	Записать в регистровую пару HL адрес ячейки памяти 0900 (0900 →HL)
0803	7E		MOV A, M	Получить число из адреса указанного в регистровой паре HL ($M_{HL} \rightarrow A$)
0804	2F		CMA	Инвертировать число в аккумуляторе
0805	23		INX H	Увеличить на 1 число в регистровой паре HL ($HL = HL + 1$)
0806	77		MOV M, A	Записать число из аккумулятора по адресу, указанному в HL ($A \rightarrow M_{HL}$)
0807	C3 00 00		JMP Label1	Перейти на метку

Ход работы

Задание 1. Исследовать программу

Порядок выполнения:

1. Ввести в программную модель УМК 80 программу (рисунок 4).

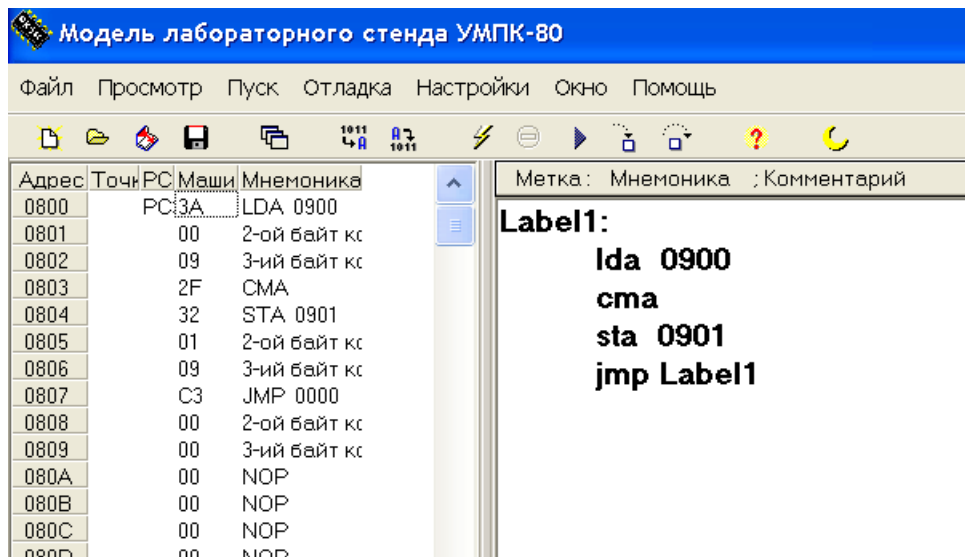


Рисунок 4 – Пример ввода программы

2. Нажать кнопку «Ассемблирование команды».
 3. Записать по адресу 0900 исследуемое число.
 4. Осуществить пуск программы. Проверить результат выполнения программы путем считывания результата из ячейки 0901.
 5. Исследовать процесс выполнения программы по командным циклам, последовательно нажимая кнопку «Шаг команды», выполнить программу. Проанализировать, как осуществляется обход программы и показания индикаторов ША, ШД в окне «Средства стенда» и регистра состояния на каждом шаге в окне «Регистры и флаги».
 6. Исследовать процесс выполнения программы 1.3 по машинным циклам. Обратит внимание на последовательность передачи и преобразования информации в микро-ЭВМ при выполнении каждой команды.
 7. В программу 1.3 при запуске ввести адрес точки останова. Например, адрес 0803. В точках останова исследовать содержимое аккумулятора и ячейки памяти 0901.
 8. Заменить в программе 1.3 команду JMP Label1 на команду HLT. Запустить программу, исследовать содержимое регистров МП и ячейки 0901. Сравните содержимое аккумулятора до, и после замены указанных команд.
 9. Последовательно замените в программе 1.3 команду CMA на команды INR A (3C), DCR A (3D), ADD A (87), ANA A (A7), ORA A (B7), CMP A (BF), DAA (27). Исследуйте результат выполнения программы по числу, записанному в ячейке 0901.
- Данные сведите в таблицу 20.

Таблица 20 – таблица результата выполнения программы

Число, записанное по адресу 0900	Команда, записанная по адресу 0803	Развернутое назначение команд	Число, записанное по адресу 0901

Задание 2. Исследовать программу

Порядок выполнения:

1. Ввести в программную модель УМПК 80 программу (рисунок 5).

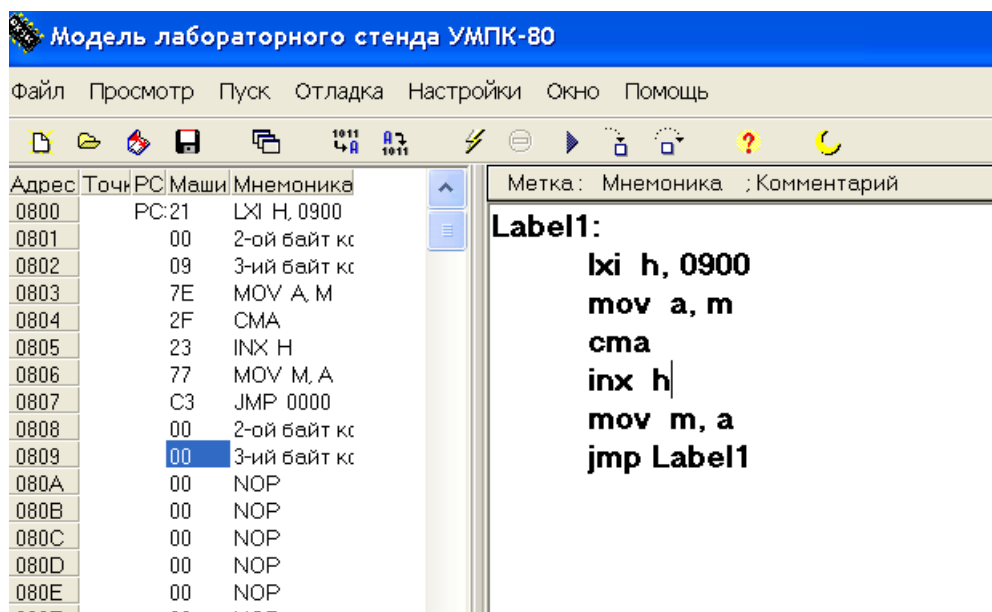


Рисунок 5 – Пример ввода программы 2.1

2. Записать по адресу 0900 исследуемое число.
3. Осуществить пуск программы с адреса 0800. Проверить результат выполнения программы по числу, записанному по адресу 0901.
4. Исследовать процесс выполнения команды MOV A, M по машинным циклам.

Содержание отчета

Отчет должен содержать:

1. Титульный лист с выходными данными лабораторной работы.
2. Программы 1.3 и 2 со словесными комментариями того, что эти программы делают.
3. Ответы на вопрос что такое прямая и косвенная адресация.
4. Анализ действий проведенных в п. 4 – 7 задания 1.
5. Таблицу, заполненную по результатам выполнения п. 8 задания 1. Под таблицей расположить ручной проверочный расчет всех пунктов таблицы в двоичной форме (вычитание через дополнительный код).
6. Результаты анализа проведенного после выполнения п. 4 задания 2.
7. Разработанную в процессе домашней подготовки программу (задается преподавателем).

Задание для домашней подготовки

Разработайте программу:

- 1) увеличения на 05h числа, записанного по адресу 0900, и записи результата по адресу 0901. Использовать косвенный способ адресации;
- 2) увеличения на 0Ah числа, записанного по адресу 0900, и записи результата по адресу 0903. Использовать прямой способ адресации;
- 3) сложения чисел, записанных по адресам 0900 и 09A0, и записи результата по адресу 0901. Использовать косвенный способ адресации;
- 4) сложения чисел, записанных по адресам 0900 и 0904, и записи результата по адресу 0903. Использовать прямой способ адресации;

- 5) вычислить разность чисел, записанных по адресам 0900 и 09A0, и записи результата по адресу 0901. Использовать косвенный способ адресации;
- 6) вычислить разность чисел, записанных по адресам 0902 и 09A0, записать результата по адресу 0901. Использовать прямой способ адресации;
- 7) уменьшения на 07h числа, записанного по адресу 0900, и записи результата по адресу 0901. Использовать косвенный способ адресации;
- 8) уменьшения на 0Ch числа, записанного по адресу 0902, и записи результата по адресу 0901. Использовать прямой способ адресации;
- 9) осуществляющую логическую операцию И над двумя числами, расположенными по адресам 0900 и 0901. Результат занести по адресу 0902. Использовать косвенный способ адресации;
- 10) осуществляющую логическую операцию И над двумя числами, расположенными по адресам 0900 и 0901. Результат занести по адресу 0902. Использовать прямой способ адресации;
- 11) осуществляющую логическую операцию ИЛИ над двумя числами, расположенными по адресам 0900 и 0901. Результат занести по адресу 0902. Использовать косвенный способ адресации;
- 12) осуществляющую логическую операцию ИЛИ над двумя числами, расположенными по адресам 0901 и 0902. Результат занести по адресу 0900. Использовать прямой способ адресации;
- 13) осуществляющую логическую операцию исключающее ИЛИ над двумя числами, расположенными по адресам 0900 и 0901. Результат занести по адресу 0902. Использовать косвенный способ адресации;
- 14) осуществляющую логическую операцию исключающее ИЛИ над двумя числами, расположенными по адресам 0900 и 0901. Результат занести по адресу 0902. Использовать прямой способ адресации;
- 15) увеличения на DDh числа, записанного по адресу 0901, и записи результата по адресу 0902. Использовать косвенный способ адресации;

16) уменьшения на 07h числа, записанного по адресу 0901, и записи результата по адресу 0900. Использовать косвенный способ адресации;

17) осуществляющую логическую операцию И над двумя числами, расположенными по адресам 0902 и 0901. Результат занести по адресу 0900. Использовать косвенный способ адресации;

18) осуществляющую логическую операцию ИЛИ над двумя числами, расположенными по адресам 0902 и 0901. Результат занести по адресу 0900. Использовать косвенный способ адресации;

19) осуществляющую логическую операцию исключающее ИЛИ над двумя числами, расположенными по адресам 0902 и 0901. Результат занести по адресу 0900. Использовать косвенный способ адресации;

20) сложения чисел, записанных по адресам 0900 и 0901, и записи результата по адресу 0902. Использовать косвенный способ адресации;

21) вычислить разность чисел, записанных по адресам 0900 и 0901, и записи результата по адресу 0902. Использовать косвенный способ адресации;

22) вычислить разность чисел, записанных по адресам 0902 и 0901, и записи результата по адресу 0900. Использовать косвенный способ адресации.

2.4.3 Лабораторная работа № 3 «Изучение выполнения команд ввода и вывода информации»

Цель работы: исследование организации обмена информацией с устройствами ввода-вывода. Изучение программных способов маскирования данных, организации условных переходов и программных временных задержек в микро-ЭВМ.

Оборудование: персональный компьютер, программная модель станда УМПК-80.

Краткие сведения из теории

В системе команд МП КР580ВМ80А имеются команды **IN port** и **OUT port** предназначенные для чтения и записи данных во внешние устройства (ВУ). Эти команды работают только с аккумулятором. (А). Это означает, что при использовании команды **OUT** данные посылаются во внешнее устройство из аккумулятора, а при применении команды **IN** считываются из ВУ в аккумулятор. Адрес внешнего устройства (**port**) указывается непосредственно в команде. Размерность этого адреса составляет один байт и благодаря этому МП с помощью команд **IN** и **OUT** может обращаться не более чем к 256 ВУ.

В качестве ВУ могут применяться различные микросхемы – параллельные и последовательные порты, таймеры, контроллеры прерывания и т. п.

В данной лабораторной работе в качестве ВУ используется программируемый параллельный порт (ППП) типа КР580ВВ55.

В таблице 21 перечислены основные операции, реализуемые этой микросхеме.



Рисунок 6 – Формат УСРР порта КР580ВВ55А

Таблица 21 – основные операции выполняемые над микросхемой порта

Операция	Сигналы управления					Операции
	A1	A0	RD	WR	CS	
Чтение	0	0	0	1	0	Порт А → ШД
	0	1	0	1	0	Порт В → ШД
	1	0	0	1	0	Порт С → ШД
Запись	0	0	1	0	0	ШД → Порт А
	0	1	1	0	0	ШД → Порт В
	1	0	1	0	0	ШД → Порт С
	1	1	1	0	0	ШД → РУС
Отключение	X	X	X	X	1	ШД и порты в высокоомном состоянии

В состав микросхемы входят три порта, функциональное назначение которых определяется кодом управляющего слова режима работы (УСРР). УСРР загружается во внутренний регистр ППП – регистр управляющего слова (РУС). Сигналы на адресных входах А0 и А1 ППП производят селекцию одного из трех портов А, В, С или РУС. Причем порты доступны как по чтению, так и по записи, а РУС только по записи. Как видно из таблицы 1 самый младший адрес имеет порт А, затем идет порт В, затем порт С и наконец самый старший адрес соответствует РУС.

Порт может программно устанавливаться (в соответствии с кодом УСРР) в три режима (биты 5 и 6 УСРР) (рисунок 7):

- 0 – основной режим ввода-вывода информации;
- 1 – стробируемый ввод-вывод информации;
- 2 – режим двунаправленной шины.

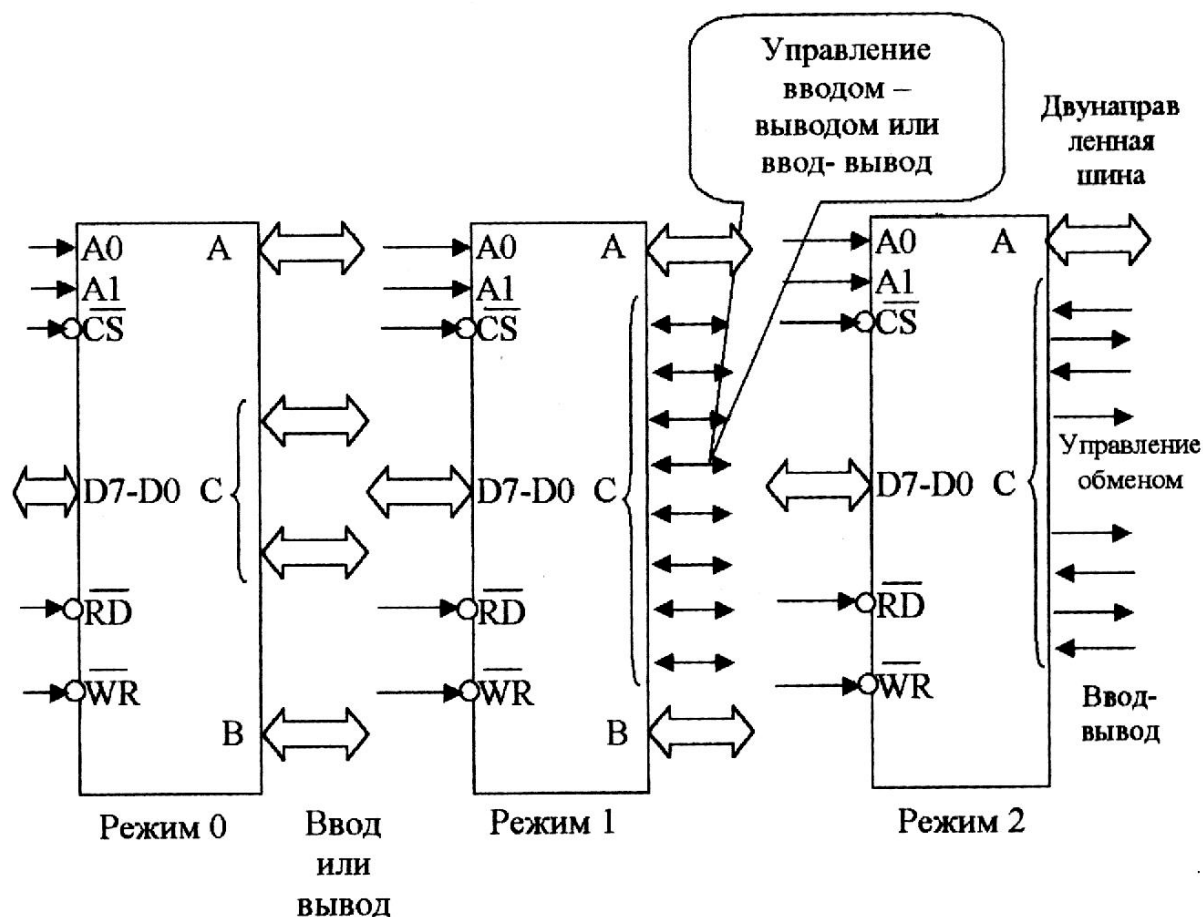


Рисунок 7 – Режимы работы порта KP580BB55A

Любой из трех перечисленных режимов может быть выбран в ходе выполнения программы и установлен загрузкой кода УСРР по команде **OUT port**.

В данной лабораторной работе параллельный порт находится внутри УМК и к нему подключены клавиатура и дисплей, расположенные на лицевой панели отладочного устройства. Доступ к ним возможен во время выполнения программы пользователя.

Микросхема порта в УМК работает всегда в нулевом режиме.

Адреса портов А, В, С и РУС лежат в диапазоне F8h – FBh.

Шестипозиционный дисплей на семисегментных полупроводниковых индикаторах работает в программном мультиплексном режиме, причем для выдачи байта индикации используется порт В, а для выбора позиции индикации порт А (рисунок 3.3). Выбор столбца опроса на клавиатуре осуществляется через тот же порт, через который производится выбор позиции индикации, а считывание кода нажатой клавиши производится через порт С (рисунок 3.3).

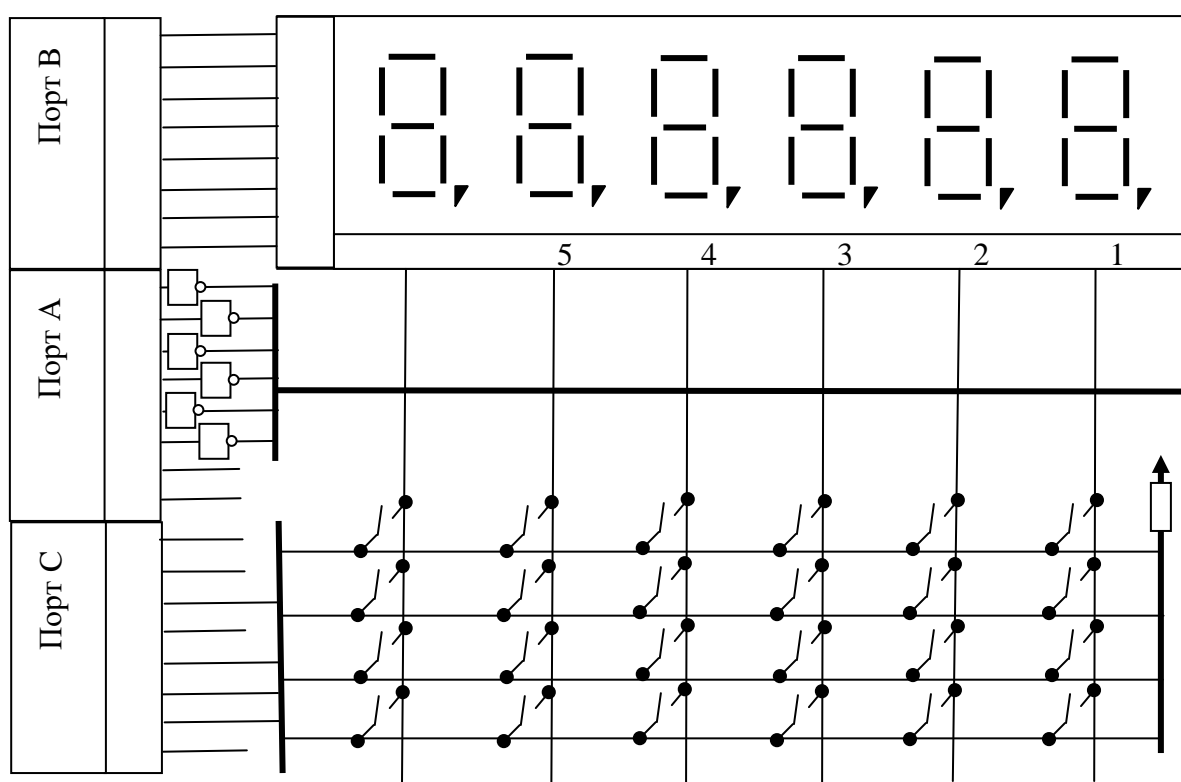


Рисунок 8 – Схема подключения клавиатуры и индикатора к параллельному порту КР580ВВ55А в УМК

Если нажатая клавиша находится в верхней строке, то считывается код 64h, если во второй сверху – 34h, в третьей сверху – 54h и, наконец, в нижней – 70h. Если не нажата ни одна клавиша, то код равен 74h. Считыванию кода отличного от указанных означает, что либо нажаты несколько клавиш, либо произошла какая – либо ошибка.

Любая программа для работы с параллельным портом состоит из следующих разделов:

- настройки микросхемы порта на режим работы;
- обмена данными с портами А, В и С.

Например:

Программа 1

Таблица 22 – программа 1 обмена данными портов А и В

Адрес	Код	Метка	Мнемокод	Комментарий
0800	3E 90		MVI A, 90	УСРР → А
0802	D3 83		OUT 83	А → РУС КР580ВВ55А
0804	DB 80	Label:1	IN 80	Порт А → А
0806	D3 81		OUT 81	А → порт В
0808	C3 0408		JMP 0804	Переход к адресу 0804 (циклическое выполнение программы)

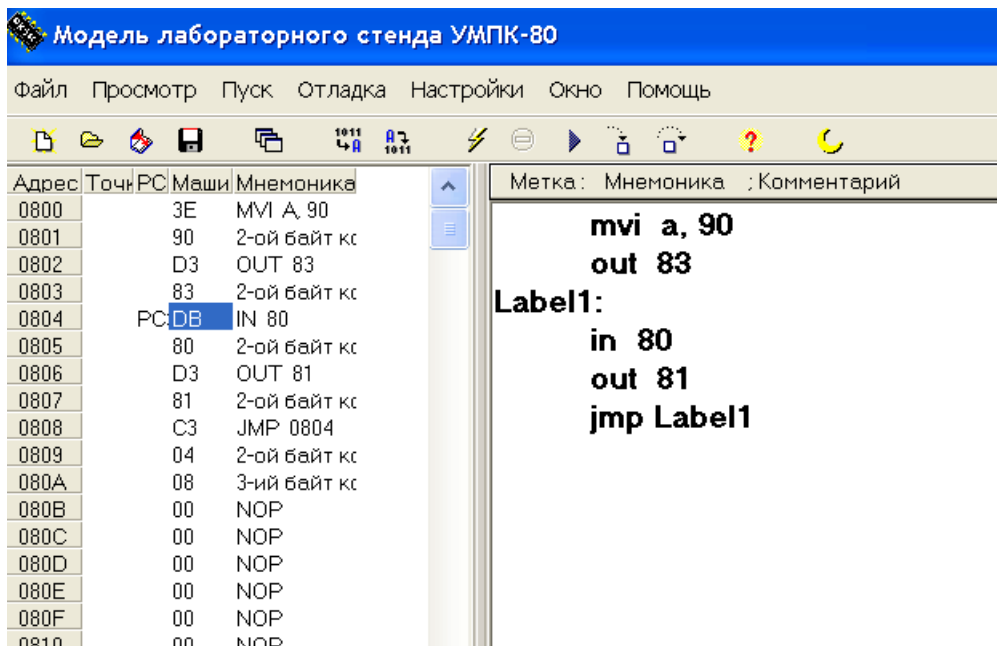


Рисунок 9 – Пример ввода программы 1

Организация условных переходов в микро-ЭВМ осуществляется с помощью регистра признаков F МП (таблица 23).

Таблица 23 – таблица регистр признаков F

7	6	5	4	3	2	1	0
S	Z	0	AC	0	P	1	CY

Регистр признаков имеет пять разрядов, каждый из которых устанавливается по определенному правилу в зависимости от результата выполнения МП последней команды.

1. В разряд переполнения **CY** записывается 1, если при выполнении арифметической команды или команды сдвига был перенос из седьмого разряда в восьмой или заем из восьмого в седьмой, в противном случае в данный разряд записывается 0.

2. В разряд **S** записывается 1, если при выполнении арифметической или логической операции получено отрицательное число, в противном случае в разряд записывается 0.

3. В разряд нулевого результата **Z** записывается 1, если при выполнении арифметической или логической команды получен нулевой результат в аккумуляторе, в противном случае в разряд записывается 0.

4. В разряд переполнения **AC** записывается 1, если при выполнении команд в аккумуляторе возникает единица переноса из третьего разряда числа в четвертый (перенос между тетрадами внутри байта).

5. В разряд четности **P** записывается 1, если при выполнении команды количество единиц в разрядах аккумулятора будет четным.

Во многих случаях при выполнении программ необходимо проверять или изменять (маскировать) состояние одного или нескольких разрядов числа в аккумуляторе (таблица 24). Это можно осуществить с помощью следующих операций:

1. Логического умножения ($\&$) числа в аккумуляторе и маски, очищает разряд числа, если в соответствующем разряде маски будет записан 0, и не изменяет его, если в разряде маски записана 1.

2. Логического сложения (\vee) числа в аккумуляторе и маски, устанавливает разряд числа в 1, если в таком же разряде маски будет записана 1, и не изменяет его, если в этом разряде записан 0.

3. Логического «исключающего ИЛИ» (\oplus) числа в аккумуляторе и маски, которое инвертирует содержимое разряда числа, если при совпадении значений разрядов числа и маски в соответствующем разряде маски записана 1, и не изменяет его, если в этом разряде записан 0.

Таблица 24 – Примеры маскирования данных

Мнемо-код	Число в аккумуляторе	Маска /байт data	Комментарий	Результат в аккумуляторе
ANI data	1111 1111	0000 0011	Логическое умножение содержимого аккумулятора с байтом data	0000 0011
	1010 1010	0110 0101		0010 0000
	0000 1111	1111 1111		0000 1111
	0101 0111	0000 0000		0000 0000
ORI data	0011 0011	0101 0101	Логическое сложение содержимого аккумулятора с байтом data	0111 0111
	0000 1111	0000 1111		0000 1111
	1111 0000	0000 1111		1111 1111
XRI data	1111 1111	1111 1111	Логическое «исключающее ИЛИ» содержимого аккумулятора с байтом data	0000 0000
	0000 1111	0000 1111		0000 0000
	1111 0000	0000 1111		1111 1111

Проведение логических операций возможно также с содержимым аккумулятора и внутренними регистрами МП. При выполнении всех логических команд задействуются разряды Z, S, P, AC регистра признаков F (в разряд CY записывается 0).

Ниже приведена программа 2 (таблица 25), (рисунок 10) выделения третьего разряда числа, считанного из порта A и записи результата маскирования в порт B.

Программа 2

Таблица 25 – программа 2

Адрес	Код	Метка	Мнемокод	Комментарий
0800	3E 90		MVI A,90	УСРР → A
0802	D3 83		OUT 83	A → РУС КР580ВВ55А
0804	DB 80	Label:1	IN 80	Порт A → A
0806	E6 08		ANI 08	A = A & 08
0808	D3 81		OUT 81	A → порт B
080A	C3 0804		JMP 0408	Выполнение программы в цикле

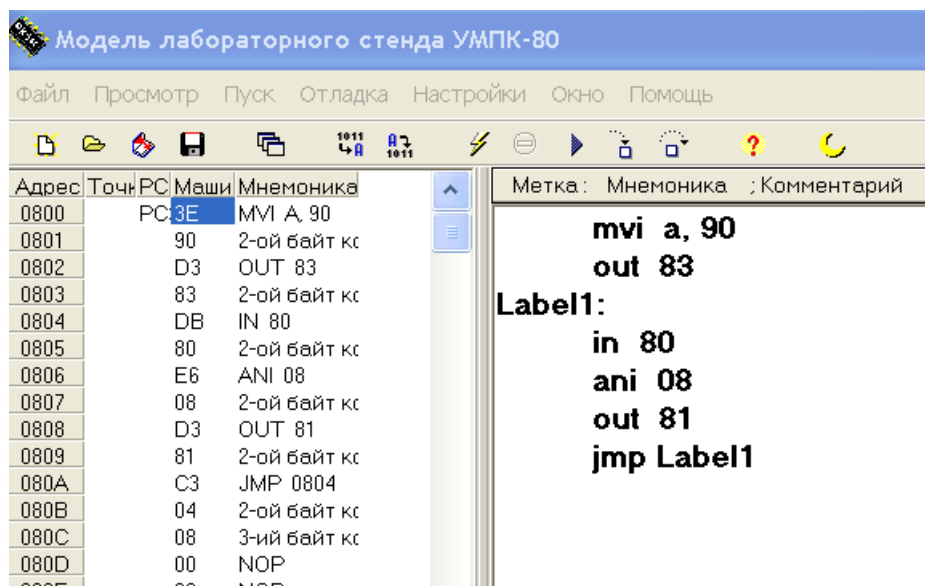


Рисунок 10 – Пример ввода программы 2

Условные переходы организуются в программах с помощью команд условных переходов. При выполнении этих команд МП проверяет состояние соответствующего разряда регистра состояния. Если при проверке состояний разряда регистра состояния условие не подтверждается, то выполняется следующая по порядку команда программы.

Все команды условных переходов трехбайтные. Первый байт содержит код команды, второй и третий байты – адрес строки перехода. Таким образом, команды условных переходов позволяют строить ветвящиеся алгоритмы и в зависимости значения результата выполнения программы переходить на различные участки программы.

В качестве примера приведена программа 3 которая при 1 в третьем разряде числа, считанного из порта А ППП записывает FFh, а в противном случае 00 h в порт В.

Программа 3

Таблица 26 – программа 3

Адрес	Код	Метка	Мнемокод	Комментарий
0800	3E 90		MVI A, 90	УСРР → А
0802	D3 83		OUT 83	А → РУС КР580ВВ55А
0804	3E 00	La- bel:1	MVI A, 00	00 → А
0806	D3 81		OUT 81	А → порт В (обнуление порта В)
0808	DB 80	La- bel:2	IN 80	Порт А → А
080A	E6 08		ANI 08	А = А & 08 (проверка состояния 3-го разряда числа из порта А)
080C	CA04 08		JZ M1	Идти на М1, если в 3 разряде был 0, (Z=1)
080F	3E FF		MVI A, FF	FF → А
0811	D3 81		OUT 81	А → порт В
0813	C3 08 08		JMP M2	Выполнение программы в цикле

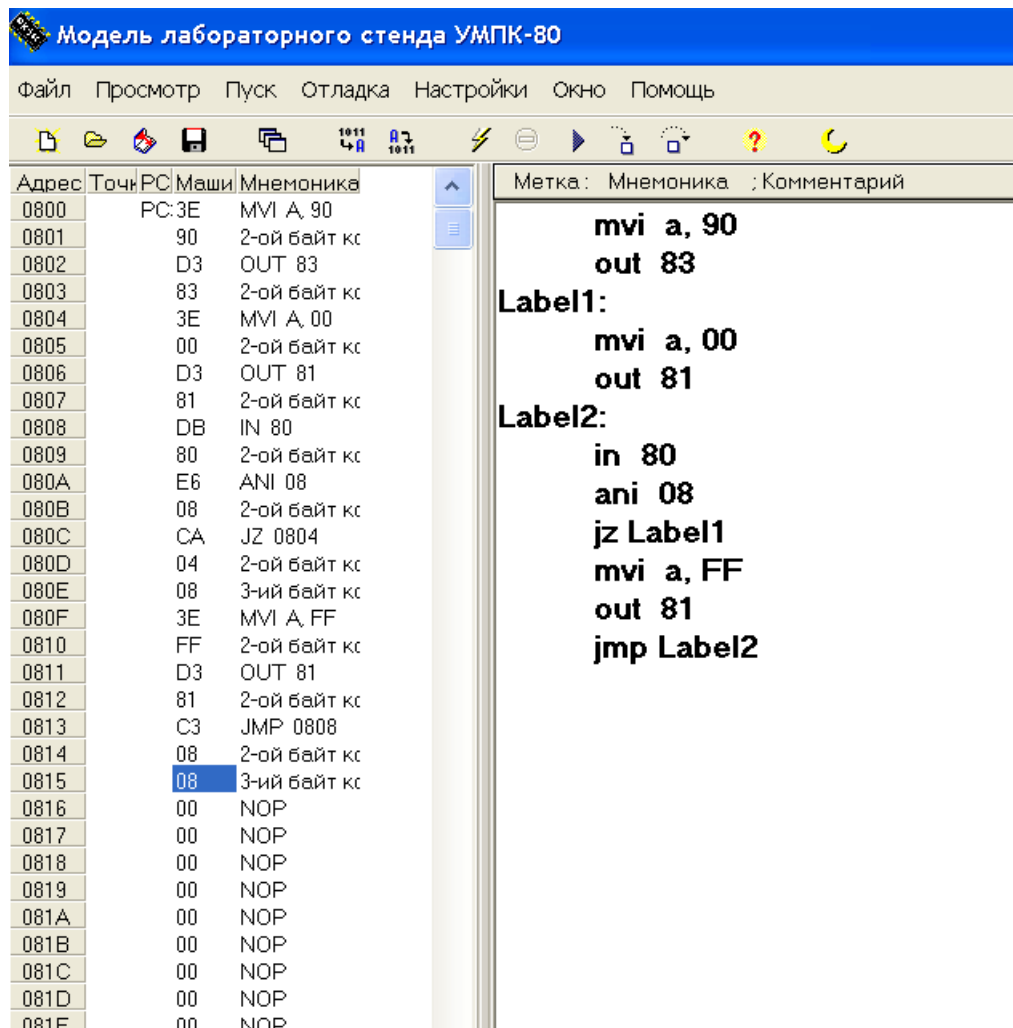


Рисунок 11 – Пример ввода программы 3

Микропроцессор КР580 позволяет организовывать программные временные задержки. Эта возможность основывается на следующем – время выполнения машинного такта для определенного изделия, выполненного на данном микропроцессоре, является фиксированной величиной. Для УМК тактовая частота МП $f = 2 \text{ МГц}$, а следовательно длительность одного машинного такта $T = 1/f = 0,5 \text{ мкс}$. Каждая команда микропроцессора выполняется за фиксированное количество машинных тактов (которое можно посмотреть в списке команд МП). Следовательно, группа команд идущих последовательно выполняется за некоторое фиксированное количество времени. Поскольку простое повторение команд занимает много места, используют циклические вычисления. Число повторов одного цикла может достигать

256 при счетчике цикла равном FFh. При необходимости увеличить временную задержку применяют вложенные циклы. Пример такой программы с двумя вложенными циклами реализующей выдержку времени порядка 0,6 с приводится ниже.

Программа 4

Таблица 27 – программа 4

Число в машинных тактов	Время выполнения команды, мкс	Метка	Мнемокод	Комментарий
7	$7 \cdot 0,5 = 3,5$		MVI B, C8	Загрузка числа повторов внешнего цикла (200) в регистр В
7	3,5	M2:	MVI DD C,	Загрузка числа повторов внутреннего цикла (221) в регистр С
4	2	M1:	NOP	Пусто
4	2		NOP	Пусто
4	2		NOP	Пусто
5	2,5		DCR C	$C = C - 1$
10	5		JNZ M1	Если не ноль идти к M1
5	2,5		DCR B	$B = B - 1$
10	5		JNZ M2	Если не ноль идти к M2

Внутренний цикл: $(2+2+2+2,5+5) \cdot 221 = 2983,5$ мкс

Всего: $3,5 + (3,5 + 2983,5 + 2,5 + 5) \cdot 200 = 3,5 + 2994,5 \cdot 200 = 598903,5$
мкс $\approx 598,9$ мс $\approx 0,599$ с

Ход работы

Внимательно прочтите теоретический материал и на его основе составьте программу, предложенную преподавателем.

Содержание отчета

Отчет должен содержать:

Титульный лист с выходными данными лабораторной работы.

Задание на программирование (выдается преподавателем).

Пояснение принципов функционирования программы и подходов к решению задачи.

Листинг программы разработанной в процессе выполнения лабораторной работы, изготовленный по общепринятым правилам.

Блок – схему программы.

Задания для лабораторной работы

1. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в крайней правой позиции дисплея. После ввода второго символа он сдвигается влево на одну позицию, а его место занимает вновь вводимый символ. При вводе третьего символа первый и второй символы вновь смещаются влево. Все это происходит до тех пор, пока первый символ не достигнет крайней левой позиции дисплея. При вводе седьмого символа

дисплей очищается, и седьмой символ должен быть выведен опять в крайней правой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

2. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в крайней левой позиции дисплея. После ввода второго символа он появляется во второй слева позиции, третий в третьей слева и т. д. до тех пор, пока не заполнятся все шесть позиций дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен опять в крайней левой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

3. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в третьей позиции дисплея (нумерация позиций начинается с нуля). Вторым символом выводится во второй позиции, третий – в четвертой, четвертый – в первой и т. д. до тех пор, пока не заполнятся все шесть позиций дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен опять в третьей позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

4. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в нулевой позиции дисплея (нумерация позиций начинается с нуля). Вторым символом выводится в пятой позиции, третий – в первой, четвертый – в четвертой и т. д. до тех пор, пока не заполнятся все шесть позиций дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен опять в нулевой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

5. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в нулевой позиции дисплея (нумерация позиций начинается

ся с нуля). Второй символ выводится во второй позиции, третий – в четвертой, четвертый – в первой, пятый – в третьей и т. д. до тех пор, пока не заполнятся все шесть позиций дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен опять в нулевой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

6. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в нулевой позиции дисплея (нумерация позиций начинается с нуля). Второй символ выводится во второй позиции, третий – в четвертой, четвертый – в пятой, пятый – в третьей и т. д. до тех пор, пока не заполнятся все шесть позиций дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен опять в нулевой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

7. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в крайней правой позиции дисплея. После ввода второго символа он сдвигается во вторую позицию, а его место занимает вновь вводимый символ. При вводе третьего символа первый и второй смещаются влево. Все это происходит до тех пор, пока первый символ не достигнет крайней левой позиции дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен в крайней левой позиции. Ввод восьмого символа смещает седьмой вправо и т. д. пока не происходит заполнение всех позиций. Далее алгоритм повторяется с начала. Программа должна работать в цикле.

8. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в крайней левой позиции дисплея. После ввода второго символа он появляется во второй слева позиции, третий в третьей слева и т. д. до тех пор, пока не заполнятся все шесть позиций дисплея. При вводе седьмого символа дисплей очищается, и седьмой символ должен быть выведен в

крайней правой позиции. Ввод восьмого символа смещает седьмой влево и т. д. пока не происходит заполнение всех позиций. Далее алгоритм повторяется с начала. Программа должна работать в цикле.

9. Организовать вывод символов на дисплей по следующему алгоритму. Четные символы, вводимые с цифровой клавиатуры (символы от 0 до F, причем 0 – четное) должны выводиться последовательно в четные позиции дисплея, начиная с нулевой. Нечетные – в нечетные позиции, начиная с первой. При заполнении всех четных позиций вывод четных цифр производить опять с нулевой позиции. При заполнении всех нечетных позиций вывод нечетных цифр производить опять с первой позиции. Программа должна работать в цикле.

10. Организовать вывод символов на дисплей по следующему алгоритму. Четные символы, вводимые с цифровой клавиатуры (символы от 0 до F, причем 0 – четное) должны выводиться последовательно в четные позиции дисплея, начиная с нулевой. Нечетные – в нечетные позиции, начиная с первой. При заполнении всех четных позиций вывод четных цифр производить в обратном порядке с четвертой позиции. При заполнении всех нечетных позиций вывод нечетных цифр производить в обратном порядке с пятой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

11. Организовать вывод символов на дисплей по следующему алгоритму. Четные символы, вводимые с цифровой клавиатуры (символы от 0 до F, причем 0 – четное) должны выводиться последовательно в четные позиции дисплея, начиная с нулевой. Нечетные – в нечетные позиции, начиная с первой. При заполнении всех четных позиций вывод четных цифр производить в обратном порядке с четвертой позиции. При заполнении всех нечетных позиций вывод нечетных цифр производить опять с первой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

12. Организовать вывод символов на дисплей по следующему алгоритму. Четные символы, вводимые с цифровой клавиатуры (символы от 0 до F, причем 0 – четное) должны выводиться последовательно в четные позиции

дисплея, начиная с нулевой. Нечетные – в нечетные позиции, начиная с первой. При заполнении всех четных позиций вывод четных цифр производить опять с нулевой позиции. При заполнении всех нечетных позиций вывод нечетных цифр производить в обратном порядке с пятой позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

13. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в крайней левой позиции дисплея. Ввод второго символа стирает первый, и второй символ выводится уже во второй слева позиции. Далее третий в следующей позиции и так до пятого символа. После ввода шестого символа, который выводится в крайней правой позиции дисплея, должны высветиться все шесть ранее введенных символов – каждый в своей позиции. Далее алгоритм повторяется. Программа должна работать в цикле.

14. Организовать вывод символов на дисплей по следующему алгоритму. Первый символ, вводимый с цифровой клавиатуры (символы от 0 до F) должен появиться в крайней правой позиции дисплея. Ввод второго символа стирает первый, и второй символ выводится уже во второй справа позиции. Далее третий в следующей позиции и так до пятого символа. После ввода шестого символа, должны высветиться все шесть ранее введенных символов – в обратном порядке позиций (т. е. первый в крайней левой позиции, второй в следующей и т. д.). Далее алгоритм повторяется. Программа должна работать в цикле.

15. Организовать вывод символов на дисплей по следующему алгоритму. Четные символы, вводимые с цифровой клавиатуры (символы от 0 до F, причем 0 – четное) должны выводиться последовательно в четные позиции дисплея, начиная с нулевой. Нечетные – в нечетные позиции, начиная с первой. При последовательном вводе двух одинаковых символов дисплей очищается, ввод начинается сначала, причем четные числа выводятся в нечетные позиции, а нечетные – в четные. При повторном вводе подряд двух одинако-

вых символов возобновляется прежний порядок вывода. Программа должна работать в цикле.

16. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от 0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне правой, должна загораться разделительная точка. После ввода последнего шестого символа они должны отобразиться на дисплее в порядке их ввода и гореть 10 секунд. Далее алгоритм повторяется. Программа должна работать в цикле.

17. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от 0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне правой, должна загораться разделительная точка. После ввода последнего шестого символа они должны по очереди загораться, начиная с последнего шестого символа, они должны последовательно загораться каждый в своей позиции, так, чтобы было заметно, что они загораются не одновременно и остаются зажженными 5 секунд. Далее алгоритм повторяется. Программа должна работать в цикле.

18. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от 0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне правой, должна загораться разделительная точка. После ввода последнего шестого символа они должны отобразиться на дисплее в порядке их ввода и, последовательно сдвигаясь вправо постепенно исчезнуть с экрана. Этот процесс должен быть визуально заметен. Далее алгоритм повторяется. Программа должна работать в цикле.

19. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от

0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне правой, должна загораться разделительная точка. После ввода последнего шестого символа они должны отобразиться на дисплее в порядке обратном их вводу и постепенно исчезнуть с экрана в следующем порядке. Сначала крайний правый символ, затем крайний левый, затем второй справа и т. д. Этот процесс должен быть визуально заметен. Далее алгоритм повторяется. Программа должна работать в цикле.

20. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от 0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне левой, должна загораться разделительная точка. После ввода последнего шестого символа они должны отобразиться на дисплее в порядке их ввода и постепенно исчезнуть с экрана в следующем порядке. Сначала символ в третьей позиции, затем символ во второй позиции, затем в четвертой позиции и т. д. Этот процесс должен быть визуально заметен. Далее алгоритм повторяется. Программа должна работать в цикле.

21. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от 0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне правой, должна загораться разделительная точка. После ввода последнего шестого символа они должны начать появляться на дисплее в порядке обратном их вводу и полностью появиться на экране в следующем порядке. Сначала крайний правый символ, затем крайний левый, затем второй справа и т. д. Этот процесс должен быть визуально заметен. Когда экран заполнится, он должен гореть 5 секунд. Далее алгоритм повторяется. Программа должна работать в цикле.

22. Организовать вывод символов на дисплей по следующему алгоритму. С клавиатуры последовательно вводятся шесть символов (символы от 0 до F), которые не отображаются на дисплее. Вместо этого, при вводе каждого символа, в следующей позиции дисплея, начиная с крайне правой, должна загораться разделительная точка. После ввода последнего шестого символа они должны отобразиться на дисплее в порядке их ввода в следующем порядке. Сначала крайний правый символ, затем крайний левый, затем второй справа и т. д. Этот процесс должен быть визуально заметен. Когда экран заполнится, он должен гореть 5 секунд. Далее алгоритм повторяется. Программа должна работать в цикле.

ЗАКЛЮЧЕНИЕ

Разработка комплекса лабораторных работ по дисциплине «Микропроцессорная техника» включает в себя:

курс теории, где рассмотрены основные вопросы по изучению материалов по дисциплине «Микропроцессорная техника»;

методику проведения лабораторных работ;

разработка тем лабораторных работ;

комплекс лабораторных работ по следующим темам.

Комплекс лабораторных работ по дисциплине «Микропроцессорная техника» должен стать полезным и необходимым инструментом для преподавателя и студентов, изучающих дисциплину «Микропроцессорная техника» на кафедре информационных систем и технологий. Подобная работа на сегодняшний день становится особенно востребованной для дистанционного обучения, что определяет значение выполненного комплекса.

Таким образом, с помощью комплекса студенты заочного обучения получают, как теоретические, так и практические знания. Комплекс позволяет повысить и усовершенствовать свой уровень подготовки по дисциплине «Микропроцессорная техника», качественно и результативно выполнять необходимые действия и проверять правильность их выполнения.

Излагаемый материал будет одинаково интересен как начинающим, изучать дисциплину «Микропроцессорная техника», так и тем, кто уже имеет определенные навыки работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ ИНФОРМАЦИИ

1. **Балашов Е. П.** Микропроцессоры и микропроцессорные системы. [Текст] : учебное пособие для вузов / Балашов Е. П., Пузанков Д. В. Под ред. Смолова В. Б. – М. : Радио и связь, 1981. – 328 с.

2. **Бирюков С.** Цифровые устройства на МОП-интегральных микросхемах. [Текст] : справочник / Бирюков С. – М. : Радио и связь, 1996. – 192 с.

3. **Волков В. Л.** МикроЭВМ : В 8 кн. [Текст] : практ. пособие / Под ред. Л. Н. Преснухина. Кн. 7. Учебные стенды / Ю. И. Волков, В. Л. Грбунов, Д. И. Панфилов, С. Г. Шаронин. – М. : Высш. Шк., 1988. – 224 с. : ил.

4. **Гришин Ю. П.** Микропроцессоры в радиотехнических системах [Текст] : / Ю.П. Гришин, Ю. М. Казаринов, В. М. Катиков – М. : Радио и связь, 1982. – 280 с., ил.

5. **Гук М.** Аппаратные средства IBM PC. [Текст]: справочник / Гук М. – СПб: Питер, 1996. – 224 с.

6. **Злобин В. К.** Программирование арифметических операций в микропроцессорах. [Текст] : учебное пособие для технических вузов / Злобин В. К., Григорьев В. Л. – М. : Высш. шк., 1991, – 303 с.

7. **Каган Б. М.** Основы проектирования микропроцессорных устройств автоматики [Текст] / Б.М. Каган, В.В. Сташин. – М.: Энергоатомиздат, 1987. –304 с.

8. **Калабеков В.** Микропроцессоры и их применение в системах передачи и обработки сигналов [Текст] : учебное пособие для вузов / В. Калабеков. – М. : Радио и связь, 1988. – 368 с.

9. **Майоров В. Г.** Практический курс программирования микропроцессорных систем. [Текст] : учебное пособие для технических вузов / Майоров В.Г., Гаврилов И. – М. : Машиностроение, 1989. – 279 с.

10. **Майоров С.** Введение в микроЭВМ [Текст] / С. Майоров, В.В. Кириллов, Приблуда. Л. : Машиностроение, 1988. – 304 с.

11. **Мячев** Персональные ЭВМ и микроЭВМ. Основы организации. [Текст] : справочник / Мячев , Степанов В. М. – М. : Радио и связь, 1991. – 238 с.

12. **Новиков Ю. В.** Основы микропроцессорной техники [Текст] : учебное пособие для вузов. / Ю.В. Новиков, П.К. Скоробогатов. – М.: Бином, 2009. – 357 с.

13. **Преснухин Л. Н.** Микропроцессоры [Текст] : учеб. для вузов. В 3 кн. Кн. 1. Архитектура и проектирование микроЭВМ. Организация вычислительных процессоров. / П. В. Нестеров, В. Ф.Шаньгин, В. Л. Горбунов и др.; под ред. Л. Н. Преснухина. –М.: Высш. шк., 1986. – 495с.

14. **Преснухин Л. Н.** Микропроцессоры [Текст] : учеб. для вузов. В 3 кн. Кн. 2. Средства сопряжения. Контролирующие и информационно-управляющие системы. / В. Д. Вернер, Н. В. Воробьев, В. Горячев и др.; под ред. Л. Н. Преснухина. – М. : Высш. шк., 1986. – 383 с.

15. **Преснухин Л. Н.** Микропроцессоры [Текст] : учеб. для вузов. В 3 кн. Кн. 3. Средства отладки, лабораторный практикум и задачник. / Н.В. Воробьев, В.Л. Горбунов, В. Горячев и др.; под ред. Л.Н. Преснухина. – М. : Высш. шк., 1986. – 351с.

16. **Пухальский Г. И.** Цифровые устройства. [Текст] : учебное пособие для втузов / Пухальский Г.И., Новосельцева Т.Я. – СПб.: Политехника, 1996. – 885 с.

17. **Пухальский Г. И.** Проектирование микропроцессорных устройств [Текст] : учебное пособие для вузов. – СПб. : Политехника, 2001. – 544 с. : ил.

18. **Ратч Э.** IBM AT. Руководство для начинающих. [Текст] : учебное пособие / Ратч Э. – М. : Радио и связь, 1993. – 263 с.
19. **Скэнлон Л.** Персональные ЭВМ IBM PC и XT. Программирование на языке ассемблера [Текст] : учебное пособие / Скэнлон Л. – М. : Радио и связь, 1989. – 254 с.
20. **Смирнов С.** Педагогика. Педагогические теории, системы, технологии. [Текст] : учебное пособие / Смирнов С. – М., 1999. – 512 с.
21. **Сташин В. В.** Проектирование цифровых устройств на однокристалльных микроконтроллерах. [Текст] : учебное пособие / В. В. Сташин, В. Урусов, О. Ф. Мологонцева, – М. : Энергоатомиздат, 1990, – 224 с.
22. **Суворов Д. Н.** Микропроцессоры и микроЭВМ в АСУ ТП предприятий строительной индустрии. [Текст] : учебное пособие / Суворов Д. Н. – МАДИ. – М., 1987. – 84 с.
23. **Суворов Д. Н.** Структура и устройство микропроцессорных систем управления. [Текст] : учебное пособие / Суворов Д. Н. – МАДИ. – М.: 1998. – 103 с.
24. **Тули М.** Справочное пособие по цифровой электронике. [Текст] : справочник / Тули М. – М. : Энергоатомиздат, 1990. – 137 с.
25. **Чернышев Ю.** Электронная и микропроцессорная техника. [Текст] : учебное пособие / Ю. Чернышев, Е. Шутов. – Томск : Изд-во Томского политехнического университета, 2010. – 135 с.
26. **Шалунова М. Г.** Практикум по методике профессионального обучения [Текст] : учебное пособие / Шалунова М. Г., Эрганова Н. Е. – Екатеринбург: Издательство Уральского государственного профессионального – педагогического университета, 2001. – 67 с.
27. **Шевкопляс Б. В.** Микропроцессорные структуры. Инженерные решения [Текст] : справочник / Шевкопляс Б. В. – М. : Радио и связь, 1993. – 256 с.

28. **Щелкунов Н. Н.** Микропроцессорные средства и системы [Текст] / Н. Н. Щелкунов, П. Дианов. – М. : Радио и связь, 1988. – 288 с.

29. **Элфринг Г.** Программирование на языке ассемблера для микроЭВМ. [Текст] : учебное пособие / Элфринг Г.– М. : Радио и связь, 1987 – 168 с.

30. **Эрганова Н. Е.** Основы методики профессионального обучения. [Текст] : учебное пособие / Эрганова Н. Е. – Екатеринбург : Изд-во Уральского государственного профессионально – педагогического университета, 1999. – 138 с.

**Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический
университет»**

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
специальность 050501 Профессиональное обучение (по отраслям)
специализация «Вычислительная техника»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Н. С. Толстова
«_____» ____ 2016 г.

**ЗАДАНИЕ
на выполнение выпускной квалификационной работы
специалиста**

Студент 5 курса, группы ВТ-503 Михалев Кирилл Сергеевич

1. Тема: Лабораторный практикум по дисциплине «Микропроцессорная техника» утверждена распоряжением по институту от 28.03.2016 г. № 57.

2. Руководитель Мешков Владислав Витальевич, старший преподаватель кафедры ИС

3. Место преддипломной практики: кафедра информационных систем и технологий

4. Исходные данные к ВКР:
Гордиевских В.М., Петухов Д.В. Учебное пособие «Технические средства обучения»

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов):

- 1) Анализ учебно-методической документации
- 2) Разработка лабораторных работ

6. Перечень демонстрационных материалов
Презентация, созданная в Power Point 2013

