

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

СОГЛАСОВАНО

Заведующий кафедрой
«Информационные технологии и
компьютерные системы»

 А.А. Брюховецкий

« 14 » апрель 2023 г.

УТВЕРЖДАЮ

Директор института
информационных технологий

 В.Н. Бондарев

« 14 » апрель 2023 г.

М.П.

РАБОЧАЯ ПРОГРАММА ПРАКТИКИ

Производственная практика

Б2.О.03(П)_Технологическая (проектно-технологическая) практика

09.03.01 Информатика и вычислительная техника

(код и направление подготовки / специальность)

Электронно-вычислительные машины, системы и сети

(наименование профиля подготовки / специализации)

бакалавриат

(уровень высшего образования)

очная, 2023

(форма обучения, год набора)

Севастополь
2023 г.

Рабочая программа практики составлена на основе ФГОС по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации № 929 от 19.09.2017.

Рабочая программа практики рассмотрена и одобрена на заседании кафедры «Информационные технологии и компьютерные системы» 17 апреля 2023 г, протокол №9.

Руководитель образовательной программы


(подпись)

А.А. Брюховецкий
(И.О. Фамилия)

Рабочая программа практики
составлена:

доцент, к.т.н.
(должность, ученая степень, ученое звание)


(подпись)

И.А. Балакирева
(И.О. Фамилия)

Рецензент:
Генеральный директор ООО
«ЛАНКОМ»


(подпись)

П.С. Никифоров
(И.О. Фамилия)

Содержание

1. Цели практики	4
2. Задачи практики	4
3. Место практики в структуре ООП	5
4. Тип практики, способ и форма ее проведения.....	5
5. Планируемые результаты обучения при прохождении практики, соотнесенные с планируемыми результатами освоения образовательной программы	5
6. Структура и содержание практики.....	8
7. Формы отчетности по практике	8
8. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по практике	10
9. Учебно-методическое и информационное обеспечение практики.....	12
10. Материально-техническое обеспечение практики	13
11. Бланки документов на практику.....	13
Приложение А.....	23

1. Цели практики

Цель практики – закрепить и расширить теоретические и практические знания, полученные в процессе обучения, а также осуществить сбор, систематизацию и обобщение материалов для выпускной квалификационной работы по следующим направлениям профессиональной деятельности:

разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием;

осваивать методики использования программных средств для решения практических задач;

разрабатывать интерфейсы "человек - электронно-вычислительная машина";

разрабатывать модели компонентов информационных систем, включая модели баз данных;

- производственно-технологическая деятельность:

разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования;

- научно-исследовательская деятельность:

обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности;

готовить презентации, научно-технические отчеты по результатам выполненной работы, оформлять результаты исследований в виде статей и докладов на научно-технических конференциях.

2. Задачи практики

Основными задачами практики являются:

- углубление и закрепление полученных теоретических знаний;
- изучение технологии применения вычислительной техники в автоматизированных системах управления на предприятии;
- изучения технологии построения, конфигурирования и администрирования компьютерных сетей на предприятии;
- исследование возможности использования на предприятии автоматизированных систем обработки информации и управления для автоматизации организационно-распорядительной деятельности, технологического процесса, складского учета, а также систем автоматизированного проектирования и ведения документации;
- экономическое обоснование необходимости внедрения автоматизированных систем управления и компьютерных сетей на предприятии;
- освоение технологического оборудования, средств вычислительной техники, контрольно-измерительных приборов и другого оборудования, приборов, средств;
- автоматизация документооборота предприятия с использованием технологий WEB-программирования.

3. Место практики в структуре ООП

Уровень высшего образования – бакалавр.

Место практики в структуре ООП – в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника, практика входит в Блок 2 «Практики» и относится к обязательной части основной образовательной программы.

Практика базируется на следующих учебных дисциплинах:

- Системное программирование;
- Компьютерная схемотехника;

- Сети и телекоммуникации.

Изучение этих дисциплин позволяет студентам, в результате успешного усвоения теоретических курсов, иметь знания, умения и компетенции для освоения программы практики.

В результате полученных в ходе выполнения заданий по практике студент получает знания, умения и навыки необходимые для выполнения и защиты ВКР.

4. Тип практики, способ и форма ее проведения

Тип практики - Технологическая (проектно-технологическая) практика.

Вид практики – производственная.

Способ проведения практики – стационарная.

Форма проведения практики – концентрированная.

5. Планируемые результаты обучения при прохождении практики, соотнесенные с планируемыми результатами освоения образовательной программы

Формируемые в результате освоения ООП компетенции	Планируемые результаты обучения при прохождении практики
ОПК-1 Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-1.1 Владеет: навыками моделирования, экспериментальный исследований ОПК-1.2 Знает: основные положения математического анализа, моделирования ОПК-1.3 Умеет: создать модель сложной системы и оценить ее характеристики, провести модельный эксперимент
ПК-1 Способен к концептуальному, функциональному и логическому проектированию систем среднего и крупного масштаба и сложности	ПК-1.1 Владеет: методами концептуального, функционального и логического проектирования систем ПК-1.2 Знает: теоретические основы моделирования ПК-1.3 Умеет: строить модели систем различного вида
ПК-2 Способен разрабатывать технические спецификации на программные компоненты, базы данных, технические системы и их взаимодействие	ПК-2.1 Владеет: навыками разработки технических спецификаций программных продуктов ПК-2.2 Знает: теоретические основы разработки спецификаций программных продуктов ПК-2.3 Умеет: разрабатывать спецификации программных продуктов
ПК-3 Способен проектировать программное и аппаратное обеспечение вычислительных систем	ПК-3.1 Владеет: навыками проектирования программного и аппаратного обеспечения вычислительных систем ПК-3.2 Знает: теоретические основы проектирования программного и

	аппаратного обеспечения вычислительных систем ПК-3.3 Умеет: проектировать программное и аппаратное обеспечение вычислительных систем
ПК-4 Способен к созданию инструментальных средств системных программных продуктов	ПК-4.1 Владеет: навыками создания инструментальных средств системных программных продуктов ПК-4.2 Знает: теоретические основы создания инструментальных средств системных программных продуктов ПК-4.3 Умеет: создавать инструментальных средств системных программных продуктов
ПК-5 Способен к созданию, администрированию и обеспечению информационной безопасности БД	ПК-5.1 Владеет: навыками создания, администрирования и обеспечения информационной безопасности БД ПК-5.2 Знает: теоретические основы создания, администрирования и обеспечения информационной безопасности БД ПК-5.3 Умеет: обеспечивать информационную безопасность БД
ПК-6 Способностью готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях	ПК-6.1 Владеет: навыками подготовки презентаций и научно-технических отчетов по результатам выполненной работы ПК-6.2 Знает: как публиковать результаты исследований в виде статей и докладов на защите результатов практики ПК-6.3 Умеет: готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на защите результатов практики

6. Структура и содержание практики

6.1 Структура практики

Общая трудоемкость практики составляет 3 зачетных единицы (108 часов), 2 недели.

№раздела	Наименование раздела практики	Виды учебной нагрузки и их трудоемкость, часы			
		Теоретические занятия	Практические занятия	СРС	Всего часов
1	Подготовительный этап.	2			2
2	Изучение литературы и подготовка обзора (реферат) по теме выпускной квалификационной работы бакалавра			28	28
3	Системный анализ объекта проектирования	2		20	22
4	Разработка специальных вопросов по теме ВКР			36	36
5	Подготовка отчета. Заключительный семинар.			20	20
ВСЕГО		4	-	104	108

6.2 Содержание практики

Подготовительный этап включает: организационное собрание, заполнение документов на практику, получение индивидуального задания.

Этап изучения литературы предполагает ознакомление с предметной областью выполнения ВКР и подготовки реферата.

На этапе выполнения системного анализа определяются цели проектирования и границы объекта проектирования, проводится системный анализ объекта проектирования.

На этапе разработки специальных вопросов по теме ВКР выполняется индивидуальное задание, выданное руководителем ВКР.

Заключительный семинар посвящен подведению итогов практики, определению порядка представления отчетов по практике и защиты результатов прохождения практики.

7. Формы отчетности по практике

По итогам практики студент представляет отчет.

Защита отчета по практике производится на комиссии кафедры, как правило, в последние два дня практики (не позднее установленного срока). Комиссия, после сообщения студента о результатах практики, вопросов и обсуждения, объявляет оценку.

Отчет по практике (20 - 25 стр.) включает в себя следующие разделы:

Содержание.

Введение.

1. Постановка задачи в соответствии с индивидуальным заданием.
2. Обзор литературных источников по теме проектирования.
3. Системный анализ объекта проектирования.
4. Специальные вопросы по теме проектирования.

Заключение.
Библиографический список.
Приложения.

Во введении кратко описывается предметная область объекта проектирования, определяется проблема, являющаяся источником темы ВКР и формулируется актуальность разработки.

В разделе «Постановка задачи» на основании индивидуального задания на практику, выдаваемого руководителем ВКР, формулируется постановка задачи ВКР, определяются исходные данные и конечный ожидаемый результат выполнения ВКР.

В разделе «Системный анализ объекта проектирования» формулируется конечная цель проектирования, определяются границы объекта проектирования, проводится анализ объекта проектирования на основании принципов системного подхода, формулируются функции и структура объекта проектирования.

Раздел «Специальные вопросы по теме проектирования» выполняется индивидуальное задание по теме ВКР.

Заключение констатирует решение всех задач, описанных в разделе «Постановка задачи», содержит обобщение практических результатов, изложенных в отчете.

Список использованной литературы (отражает источники, которые использовались при выполнении индивидуального задания и написании отчета по практике).

В приложения выносятся промежуточные результаты разработки проекта (таблицы, диаграммы, сканы «доски» и другие материалы, используемые в процессе реализации выбранного метода управления разработкой), текст программы.

Текст отчета по учебной практике набирается в Microsoft Word в формате А-4: шрифт TimesNewRoman – обычный, размер 14 пт; межстрочный интервал – полуторный; левое, верхнее и нижнее поля – 2,0 см; правое поле – 1,0 см; левое поле – 3 см, абзац – 1,25 см. Объем отчета должен быть 20 - 25 страниц.

Страницы отчета нумеруют арабскими цифрами, с соблюдением сквозной нумерации по всему тексту. Номер проставляется в центре нижней части листа (выравнивание от центра) без точки в конце номера. Цифровой материал должен оформляться в виде таблиц.

Студенты при прохождении практики обязаны в конце дня записывать выполненную работу в дневнике-графике.

8. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по практике

8.1. Порядок оценивания результатов прохождения практик обучающимися;

Оценка по практике выставляется на основе защиты отчета о практике и представляет дифференцированный зачет (с оценкой). При оценке результатов практики учитывается степень самостоятельности студентов и трудовая дисциплина. В таблице отражена балльно – рейтинговая система оценивания.

Таблица соответствия результатов контроля знаний по разным шкалам и критерии оценивания

Сумма баллов по 100-балльной шкале	Оценка ECTS	Критерии оценивания	Уровень компетентности	Оценка по национальной шкале		
				для экзамена, КП (КР), практики	для зачета	
90 – 100	A	Индивидуальное задание выполнено полностью. При выполнении задания студент работал творчески, инициативно, самостоятельно. Отчет оформлен грамотно. Доклад по результатам практики сделан четко и ясно.	Высокий (творческий)	отлично	зачтено	
82-89	B	Индивидуальное задание выполнено полностью. При выполнении задания студент работал творчески, инициативно, самостоятельно. Есть недочеты в оформлении отчета и/или в докладе.	Достаточный			
75-81	C	Индивидуальное задание выполнено полностью. Руководитель практики влиял на результаты выполнения задания. Есть недочеты в оформлении отчета и/или в докладе.				
69-74	D	Задание выполнено частично (70%). Есть недочеты в оформлении отчета и/или в докладе.	Средний			Удовлетворительно
60-68	E	Задание выполнено частично (50%). Есть недочеты в оформлении отчета и/или в докладе.				
35-59	FX	Задание не выполнено, отчет не оформлен.	Низкий	не удовлетворительно	не зачтено	
1-34	F	Задание не выполнено, отчет не оформлен. Студент не являлся на место прохождения практики.				

8.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций

На практику выдаются только индивидуальные задания, которые формирует руководитель ВКР. Он же оценивает работу студента во время практики и отражает это в своем отзыве и оценке. Окончательная оценка по результатам практики выставляется после защиты отчета перед комиссией.

9. Учебно-методическое и информационное обеспечение практики

9.1 Основная литература

1	Вдовин, В. М. Теория систем и системный анализ: учебник / В. М. Вдовин, Л. Е. Суркова, В. А. Валентинов. — 6-е изд., стер. — Москва: Дашков и К, 2022. — 644 с. — ISBN 978-5-394-03716-0. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/277577 (дата обращения: 18.04.2023).
2	Кутузов, О. И. Инфокоммуникационные системы и сети: учебник для вузов / О. И. Кутузов, Т. М. Татарникова, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 244 с. — ISBN 978-5-507-44763-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/242858 (дата обращения: 29.05.2023). — Режим доступа: для авториз. пользователей.
3	Кутузов, О. И. Инфокоммуникационные системы и сети: учебник для вузов / О. И. Кутузов, Т. М. Татарникова, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 244 с. — ISBN 978-5-507-44763-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/242858 (дата обращения: 29.05.2023). — Режим доступа: для авториз. пользователей.
4	Осокина, Е. Б. Микропроцессорные системы управления: учебное пособие / Е. Б. Осокина. — Владивосток: МГУ им. адм. Г.И. Невельского, 2020. — 129 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/171805 (дата обращения: 29.05.2023). — Режим доступа: для авториз. пользователей.

9.2 Дополнительная литература

1	Общие требования к текстовым документам. ГОСТ 2.105—95. ЕСКД. М.: «Стандартинформ», 2005. — 28 с.
2	ГОСТ 2.109-73 Единая система конструкторской документации (ЕСКД). Основные требования к чертежам (с Изменениями N 1-11). М.: Стандартинформ, 2011
3	ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. М.: Издательство стандартов № 2001.

9.3 Методические указания по практике

Методические указания по организации и проведению технологической (проектно-технологической) практики студентов направлений 09.03.01 — "Информатика и вычислительная техника" и 09.03.04 — «Программная инженерия» очной формы обучения / Сост. Балакирева И.А., Брюховецкий А.А., Скатков А.В. — Севастополь, 2021. — 18 с.

9.4 Информационные технологии

Перечень информационных технологий, используемых при проведении практики:

1. Инструментальные средства, обеспечивающие выполнение задания для ВКР.
2. Пакет Microsoft Office 2010/2016.

10. Материально-техническое обеспечение практики

В университете проводится в лабораториях кафедры. Компьютеры должны быть оснащены программным обеспечением, приведенным в пункте 9 настоящей программы, и иметь доступ в интернет.

Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы, с указанием перечня основного оборудования, учебно-наглядных пособий и используемого программного обеспечения	Адрес (местоположение) помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом (в случае реализации образовательной программы в сетевой форме; дополнительно указывается наименование организации, с которой заключен договор)
Учебная аудитория для проведения занятий лекционного и семинарского типа № В514. 12 посадочных мест. Используется программное обеспечение: 1. Пакет Desktop Education ALNG LipSAPk OLV5 1Y Academic Education Enterprise - включает в себя систему линейки Windows и офисный пакет семейства MS Office. 2. Антивирусное программное обеспечение Kaspersky Endpoint Security для бизнеса - Стандартный Russian Edition. Договор по предоставлению неисключительных (пользовательских прав (лицензий) на использование программного обеспечения для организации учебного процесса № 19-18/ЭА-223 от 15.01.2019	299053, г. Севастополь, ул. Университетская, 33

Студентам, проходящим практику вне стен университета, принимающая организация обеспечивает прохождение практики в соответствии с договором.

11. Бланки документов на практику

Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ДНЕВНИК ПРАКТИКИ

(вид и тип практики)

обучающегося

(фамилия, имя, отчество)

Институт

Кафедра

Уровень образования

Направление подготовки
(специальность)

Профиль (специализация)

_____ курс, группа _____

Обучающийся _____
(фамилия, имя, отчество)

Прибыл на предприятие (в организацию, учреждение)

М.П. “ ____ ” _____ 20 ____ г.

(подпись) (должность, фамилия и инициалы ответственного лица)

Отметки о проведении инструктажа

Наименование инструктажа	Дата проведения инструктажа	ФИО, должность инструктирующего	Подпись	
			Инструктирующего	Инструктируемого
Инструктаж по технике безопасности				
Инструктаж по охране труда				
Инструктаж по пожарной безопасности				
Инструктаж по правилам внутреннего распорядка				

Убыл из предприятия (организации, учреждения)

М.П. “ ____ ” _____ 20 ____ г.

(подпись) (должность, фамилия и инициалы ответственного лица)

Индивидуальное задание

[illegible]

Руководитель практики от Университета

(подпись)

(инициалы и фамилия)

Руководитель практики от профильной организации

(подпись)

(инициалы и фамилия)

«_____» _____ 20__ г.

(подпись)

(инициалы и фамилия)

(подпись)

(инициалы и фамилия)

«_____» _____ 20__ г.

Рабочие записи во время практики

[illegible]

Отзыв (характеристика)

(Фамилия, имя, отчество обучающегося)

Руководитель практики от профильной организации

(подпись)

(инициалы и фамилия)

« _____ » 20 ____ г.

Сведения об уровне освоения обучающимся компетенций

№	Формируемые компетенции	Руководитель от профильной организации	Руководитель от Университета
1.	ОК- (Наименование компетенций заполняет руководитель практики от института, в соответствии с учебным планом)	освоена / освоена частично/ не освоена	освоена / освоена частично/ не освоена
2.	ОК-	освоена / освоена частично/ не освоена	освоена / освоена частично/ не освоена
3.	ОПК -	освоена / освоена частично/ не освоена	освоена / освоена частично/ не освоена
4.	ПК -	освоена / освоена частично/ не освоена	освоена / освоена частично/ не освоена
5.	ПК -	освоена / освоена частично/ не освоена	освоена / освоена частично/ не освоена
6.	...		

Руководитель практики от Университета

(подпись)

(инициалы и фамилия)

Руководитель практики от профильной организации

(подпись)

(инициалы и фамилия)

«_____» _____ 20__ г.

Оценка результатов прохождения практики обучающимся

(заполняется руководителем практики от Университета)

(Фамилия, имя, отчество обучающегося)

Дата сдачи отчета «_____» _____ 20__ г.

Оценка: _____

Руководитель практики от Университета

(подпись)

(инициалы и фамилия)

Севастополь
2023 г.

ПРИЛОЖЕНИЕ А

Варианты задания, описанного в п. 8.2.1 – Описание структуры процессора, системы команд и способов кодирования

Структура 1 (рисунок А.1). Описание и система команд (таблица А.1)

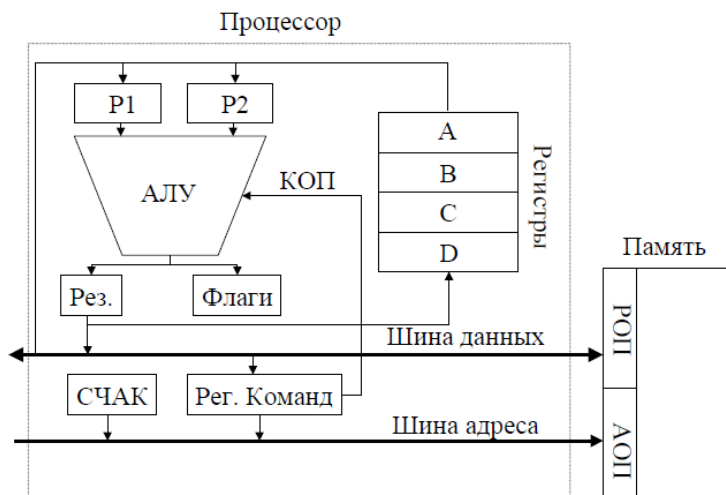


Рисунок А.1 – Структура 1

Объем оперативной памяти – 256 байт.

Длина команд – один или два байта. В первом байте биты 0..3 являются кодом операции, а следующие интерпретируются в зависимости от принадлежности команд к той или иной группе:

- для арифметических и логических команд содержат номера РОН (0-A, 1-B, 2-C, 3-D), над которыми производятся операции;
- для команд пересылки информации – номер регистра, с которым осуществляется операция;
- не используются для команд передачи управления и управления процессом.

Второй байт используется для задания адреса памяти, где находится операнд, адреса перехода для команд передачи управления или константы.

Таблица А.1 – Система команд для структуры 1

Код	Мнемокод и операнды	Описание
0000	ADD регистр1.регистр2	регистр1=регистр1+регистр2
0001	ADD регистр, память	регистр=регистр+память
0010	AND регистр1.регистр2	регистр1=регистр1&регистр2
0011	OR регистр1.регистр2	регистр1=регистр1 регистр2
0100	NOT регистр	инверсия битов регистра
0101	MOV регистр.константа	занести константу в регистр
0110	LOAD регистр, адрес	загрузка данных в регистр из памяти
0111	STORE регистр, адрес	запись данных из регистра в память
1000	JMP адрес	безусловный переход
1001	JZ адрес	переход, если 0
1010	JO адрес	переход, если переполнение
1011	JL адрес	переход, если меньше
1100	JNZC регистр, адрес	переход, если не ноль, декремент регистра
1101	CLF	установка регистра флагов в 0
1110	CALL адрес	переход к подпрограмме, адрес возврата в регистре D
1111	RET	возврат из подпрограммы

Структура 2 (рисунок А.2). Описание и система команд (таблица А.2)

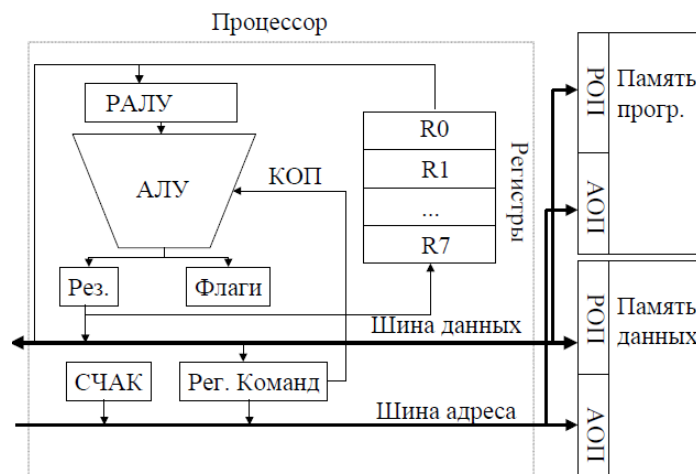


Рисунок А.2 – Структура 2

Объем оперативной памяти данных – 256 байтов, объем оперативной памяти программ – 256 байтов. Данные и программы хранятся отдельно.

Таблица А.2 – Система команд для структуры 2

Код	Мнемокод и операнды	Описание
00000	ADD POH	рез=POH+рез
00001	ADD адрес	рез=память[адрес]+рез
00010	ADC POH	рез=POH+рез+флаг переполнения
00011	SUB POH	рез=рез-POH
00100	SUB адрес	рез=рез-память[адрес]
00101	SBB POH	рез=рез-POH-флаг переполнения
00110	AND POH	рез=POH&рез
00111	AND адрес	рез=память[адрес]&рез
01000	OR POH	рез=POH рез
01001	OR адрес	рез=память[адрес] рез
01010	XOR POH	рез=POH⊗рез
01011	XOR адрес	рез=память[адрес]⊗рез
01100	MUL POH	рез=рез*POH
01101	DIV POH	рез=рез/POH (деление нацело, остаток – в POH+1)
01110	LDR POH	рез=POH
01111	SVR POH	POH=рез
10000	CALL POH,адрес	переход к подпрограмме; адрес возврата сохраняется в POHе;
10001	RET POH	возврат из подпрограммы; адрес возврата находится в POHе
10010	INC POH	увеличение POHа на 1
10011	DEC POH	уменьшение POHа на 1
10100	MOV POH,константа	занести в POH константу
10101	MOV POH,адрес	POH=память[адрес]
10110	MOV память,POH	память[адрес]=POH
10111	LOAD POH	рез=память[адрес=POH]
11000	SAVE POH	память[адрес=POH]=рез
11001	JMP адрес	безусловный переход
11010	JZ адрес	переход, если 0
11011	JNZ адрес	переход, если не 0
11110	HALT	останов
11111	CLR	установка регистра результата в 0

Команды имеют длину один или два байта. Первый байт определяет код команды в соответствии с таблицей команд (бит 0-4), а также номер регистра-операнда (биты 5-7), указывающего на один из РОНов процессора.

Второй байт определяет адрес операнда в памяти, или задает константу.

Арифметические и логические операции выполняются над операндом, находящимся в памяти или регистре, и результатом предыдущей операции, который находится в регистре результатов. Флаг переполнения учитывается как единица младшего разряда, если используется в арифметических командах.

Структура 3 (рисунок А.3). Описание и система команд (таблица А.3)

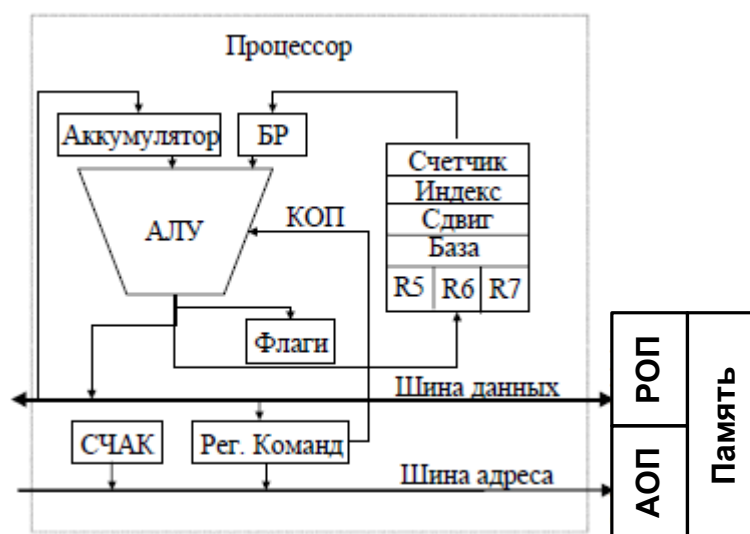


Рисунок А.3 – Структура 3

Объем оперативной памяти – 256 байт. Микропроцессор имеет 8 регистров, 5 из них – специального назначения:

РОН0 (AC). аккумулятор (в него заносятся результаты выполнения всех операций);

РОН 1 (CNT) . счетчик (для операций увеличения (уменьшения) на 1);

РОН 2 (SH) . сдвиговый регистр (для выполнения команд сдвига);

РОН 3 (BASE) . регистр базы (содержит базовый адрес операнда в ОП);

РОН 4 (IND) . регистр индекса (для формирования смещения операнда в ОП).

Таблица А.3 – Система команд для структуры 3

Код	Мнемокод	Описание
0000	ADD регистр	аккумулятор= аккумулятор + регистр
0001	ADC регистр	аккумулятор= аккумулятор + регистр + флаг переполнения
0010	SUB регистр	аккумулятор= аккумулятор - регистр
0011	SBB регистр	аккумулятор= аккумулятор - регистр - флаг переполнения
0100	AND регистр	аккумулятор=аккумулятор & регистр
0101	OR регистр	аккумулятор=аккумулятор регистр
0110	NOT	аккумулятор=не аккумулятор
0111	MOV регистр, константа	регистр=константа
1000	STOS	память[BASE+IND]= аккумулятор IND=IND+1
1000	LODS*	Аккумулятор =память[BASE+IND], IND=IND+1
1001	OUTR регистр, адрес	память[адрес]= регистр
1001	INR* регистр, адрес	регистр= память[адрес]
1010	LOOP адрес	переход по адресу, если CNT>0, CNT=CNT-1
1011	JNZ адрес	переход, если не 0
1100	JB адрес	переход, если результат меньше 0
1101	SWAP регистр	Обмен данными между аккумулятором и регистром
1110	SHR	Сдвиг вправо регистра SH на число разрядов в регистре CNT
1111	SHL	Сдвиг влево регистра SH на число разрядов в регистре CNT

находящимися в аккумуляторе и (или) регистре.

Коды команд:

Команды имеют длину один или два байта. Биты 0-3 первого байта определяют код команды в соответствии с таблицей команд. Номер регистра хранится в битах 4-6 первого байта. Бит 7 указывает направление перемещения данных: 0 означает перемещение данных в оперативную память, в остальных случаях он равен 1. Второй байт определяет константу или адреса памяти. Арифметические и логические операции выполняются над операндами,

Структура 4 (рисунок А.4). Описание и система команд (таблица А.4)

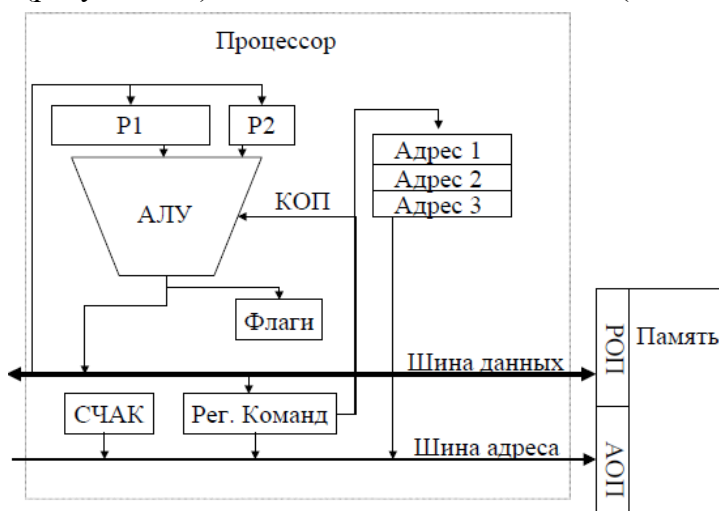


Рисунок А.4 – Структура 4

Объем оперативной памяти . 2048 байт (8 страниц по 256 байт).

Регистр СЧАК имеет 12 битов, но старший бит не используется.

Микропроцессор имеет три адресных регистра. Регистры АДРЕС1 (A1) и АДРЕС2 (A2) содержат адреса операндов команд, регистр АДРЕС3 (A3) . адрес результата операции. Эти регистры имеют по 8 битов и определяют смещение в диапазоне 0-255 внутри страницы. Номер страницы (целое число от 0 до 7) задан в битах 5-7 регистра команд.

Команды имеют длину один или два байта. Биты 0-4 первого байта определяют код команды в соответствии с таблицей команд. Биты 5-7 первого байта задают номер страницы ОП, в которой располагаются данные. Второй байт определяет константу.

Таблица А.4 – Система команд для структуры 4

Код	Мнемокод	Описание
00000	MOVA1 константа	A1=константа
00001	MOVA2 константа	A2=константа
00010	MOVA3 константа	A3=константа
00100	ADD страница*	Память[A3]= Память[A1]+ Память[A2]
00101	ADC страница	Память[A3]= Память[A1]+ Память[A2]+флаг переп.
00110	SUB страница	Память[A3]= Память[A1]- Память[A2]
00111	SBB страница	Память[A3]= Память[A1]- Память[A2] -флаг переп.
01000	MUL страница	Память[A3]= Память[A1]* Память[A2]
01001	DIV страница	Память[A3]= Память[A1]/ Память[A2]
01010	MOD страница	Память[A3]= Память[A1]% Память[A2]
01011	ABS страница	Память[A3]= Память[A1] **
01100	AND страница	Память[A3]=Память[A1]& Память[A2]
01101	OR страница	Память[A3]= Память[A1] Память[A2]
01110	XOR страница	Память[A3]= Память[A1] ^ Память[A2]
01111	NOT страница	Память[A3]=not Память[A1]
10000	JMP страница	Переход по адресу A3
10001	JB страница	Переход по адресу A3, если результат меньше 0
10010	JNZ страница	Переход по адресу A3, если результат не равен 0
10101	MOVS	Память[A3]=Память[A1], A3+=1, A1+=1
10101	CMPS	Сравнение данных Память[A1] и Память[A3]***

** отрицательные числа представляются в дополнительном коде. Для нахождения модуля необходимо выполнить перевод в прямой код и обнулить старший бит

*** выполняется аналогично операции вычитания, но результат операции не заносится в память

Структура 5 (рисунок А.5). Описание и система команд (таблица А.5)

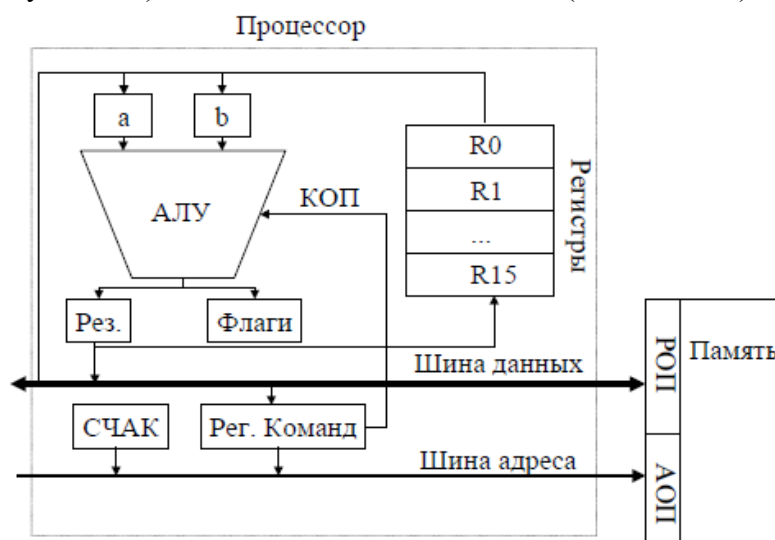


Рисунок А.5 – Структура 5

Объем оперативной памяти 256 байт. Буферные регистры а и b предназначены для хранения операндов операций. Рабочий регистр Рез используется как регистр результата, полученного при обработке данных в АЛУ.

Коды команд:

Команды имеют длину один или два байта. Первый байт определяет код команды в соответствии с таблицей команд (биты 0-3). Следующие 4 бита предназначены для указания номера регистра, коммутируемого к буферному регистру а.

Второй байт используется для задания константы или адреса памяти, где находится операнд. Если в качестве второго операнда используется регистр, то его номер хранится в младших 4-х битах второго байта команды, а старшие 4 бита не используются.

Таблица А.5 – Система команд для структуры 5

Код	Мнемокод	Описание
0000	ADNAB регистр, регистр	$\bar{a} + b$
0001	ADANB регистр, регистр	$a + \bar{b}$
0010	ADAB регистр, регистр	$a + b$
0011	ANBONAB регистр, регистр	$\bar{a}\bar{b} \vee \bar{a}b$
0100	ABONANB регистр, регистр	$ab \vee \bar{a}\bar{b}$
0101	NAB регистр, регистр	$\bar{a}\bar{b}$
0110	ANB регистр, регистр	$a\bar{b}$
0111	JZ адрес	Переход, если результат равен 0
1001	JL адрес	Переход, если результат меньше 0
1010	JMP адрес	Безусловный переход
1100	MOVC регистр, константа	Регистр:=Константа
1101	MOVR регистр, адрес	Регистр:=Память[адрес]
1110	MOVM регистр, адрес	Память[адрес]:=регистр
1111	CMP регистр, регистр	Сравнение двух регистров. Установка флагов.

Приложение Б

Детализация описания мини-проектов, представленных в п. 8.2.2

1. Калькулятор

Реализовать простейший калькулятор, который дает возможность пользователю складывать, умножать, вычитать, делить числа, находить их квадратный корень.

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я увижу сумму двух чисел

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я могу увидеть разность двух чисел

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я увижу произведение двух чисел

КАК пользователь

КОГДА я ввожу два числа

И первое число является числителем

И второе является знаменателем и *не равно* нулю

ТОГДА я увижу частное двух чисел

КАК пользователь

КОГДА я ввожу два числа

И первое число является числителем

И второе является знаменателем и *равно* нулю

ТОГДА я должен увидеть ошибку

2. Генератор паролей

Создайте генератор случайных паролей для своих друзей и семьи, чтобы обезопасить их учётные записи!

Реализовать простейший генератор паролей, который дает возможность пользователю получить пароль, удовлетворяющий его требованиям.

КАК пользователь

КОГДА я открываю программу

И указываю количество обычных символов

ТОГДА я вижу сгенерированный пароль

И этот пароль содержит выбранное количество обычных символов

КАК пользователь

КОГДА я открываю программу

И указываю количество специальных символов
ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество специальных символов

КАК пользователь
КОГДА я открываю программу
И указываю количество букв в верхнем регистре
ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество букв в верхнем регистре

КАК пользователь
КОГДА я открываю программу
И указываю различные критерии(разное число обычных, специальных, в верхнем регистре символов и цифр)
ТОГДА я вижу сгенерированный пароль
И этот пароль должен удовлетворять *всем* требованиям

3. Игра-викторина

Приложения-викторины задают пользователям серию вопросов и дают им возможность ответить. Затем викторина даёт пользователю результаты. Реализовать программу-викторину на любую интересную вам тему. Да, можно про Наруто. Да, можно не про аниме. На любую тему :3

КАК пользователь
КОГДА я открываю программу
ТОГДА программа *предлагает мне сыграть в игру (с)* и ответить на 10 вопросов

КАК пользователь
КОГДА я не соглашаюсь сыграть в игру
ТОГДА программа благодарит за участие и завершается

КАК пользователь
КОГДА я соглашаюсь сыграть в игру
ТОГДА программа задает мне вопрос и дает возможность ответить

КАК пользователь
КОГДА я даю ответ на вопрос
ТОГДА программа задает мне следующий вопрос

КАК пользователь
КОГДА я даю ответ на последний (десятый) вопрос
ТОГДА программа показывает результаты(сколько очков заработано)
И программа предлагает сыграть снова

4. Игра "Угадай число"

В данном мини-проекте вам необходимо реализовать игру, в которой компьютер пытается угадать целое число от 1 до 1000. Пользователь загадывает число, а программа задает ему вопросы из разряда *Да* или *Нет* в попытках угадать число. Если программа угадает число, она спрашивает у пользователя, это ли число она угадала. Если да - то урааа! Если нет - программа показывает грустный смайлик и предлагает новую игру. *Предполагается, что пользователь играет и отвечает на вопросы честно, а не как всегда.*

КАК пользователь

КОГДА я открываю программу

ТОГДА программа *предлагает мне сыграть в игру (с)*

КАК пользователь

КОГДА я не соглашаюсь сыграть в игру

ТОГДА программа благодарит за участие и завершается

КАК пользователь

КОГДА я соглашаюсь сыграть в игру

ТОГДА программа предлагает мне загадать число от 1 до 1000

И после этого начинает задавать вопросы из-разряда *Да/Нет*

КАК пользователь

КОГДА я даю ответ на вопрос *Да*

ТОГДА программа задает мне следующий вопрос

КАК пользователь

КОГДА я даю ответ на вопрос *Нет*

ТОГДА программа задает мне следующий вопрос

КАК пользователь

КОГДА я даю ответ на вопрос, который не является *Да* или *Нет*

ТОГДА программа просит ввести ответ снова, так как у нее лапки

КАК пользователь

КОГДА я даю ответ на вопрос

И программа предполагает что угадала число

ТОГДА программа спрашивает - такое ли число было загадано

И предлагает дать ответ на вопрос *Да* или *Нет*

КАК пользователь

КОГДА я даю ответ на вопрос

И программа угадала число

ТОГДА программа показывает результаты - как быстро она угадала, сколько шагов и какое число было загадано

И программа предлагает сыграть снова

КАК пользователь

КОГДА я даю ответ на вопрос

И программа не число

ТОГДА программа показывает грустный смайлик

И программа предлагает сыграть снова

5. Программа-шифровщик

В данном мини-проекте вам необходимо реализовать программу, которая позволяет как шифровать сообщение, так и дешифровать сообщение. Предполагается, что обладая знанием ключа/ключей, вы можете зашифровать сообщение, а ваш друг - расшифровать, и наоборот.

КАК пользователь

КОГДА я открываю программу

ТОГДА программа предлагает либо зашифровать текст, либо дешифровать текст

КАК пользователь

КОГДА я ввожу текст для шифрования

И я ввожу ключ для шифрования

ТОГДА программа показывает зашифрованный текст

КАК пользователь

КОГДА я ввожу текст для дешифрования

И я ввожу ключ для дешифрования

ТОГДА программа показывает расшифрованный текст

Приложение В

Варианты заданий для мини-проекта «Интеллектуальная матрица», представленного в п. 8.2.2

Вариант 1.

Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – float.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение суммы элементов главной диагонали матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), *i*-ый элемент которого равен количеству отрицательных элементов в *i*-ой строке матрицы (без параметров);
- 6) функция, возвращающая адрес транспонированной матрицы (без параметров).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 2. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – double.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение суммы элементов под главной диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), *i*-ый элемент которого равен сумме положительных элементов *i*-ой строки матрицы (без параметров);
- 6) функция, возвращающая адрес матрицы-произведения инкапсулированной матрицы на другую матрицу (параметр – ссылка на матрицу).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 3. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – int.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение произведения элементов над главной диагональю матрицы (без параметров);

- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), i -ый элемент которого равен произведению положительных элементов i -го столбца матрицы (без параметров);
- 6) функция, возвращающая адрес транспонированной матрицы (без параметров).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 4. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – double.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение максимального элемента под второй (не главной) диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), i -ый элемент которого равен числу положительных элементов в i -ом столбце матрицы (без параметров);
- 6) функция, возвращающая адрес матрицы-произведения инкапсулированной матрицы на другую матрицу (параметр – ссылка на матрицу).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 5. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – int.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение минимального элемента над второй (не главной) диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), i -ый элемент которого равен единице, если в i -ом столбце матрицы есть положительные элементы, и нулю, в противном случае;
- 6) функция, возвращающая адрес транспонированной матрицы (без параметров).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.