

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
Высшего профессионального образования  
«Воронежский государственный педагогический университет»

**Чулюков В.А., Джахуа Д.К., Володина Н.М.**

**ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ,  
СЕТИ И ТЕЛЕКОММУНИКАЦИИ.  
ПРАКТИКУМ**

Воронеж 2012

УДК681.3-523.8(075.8)

Печатается по решению учебно-методического совета ВГПУ

Чулюков В.А., Джахуа Д.К., Володина Н.М. Вычислительные системы, сети и телекоммуникации. Практикум: Учебное пособие. – Воронеж: ВГПУ, 2012. – 73 с.

Пособие содержит необходимый минимум теоретического материала, примеры, методические рекомендации для студентов и преподавателей, варианты заданий для выполнения 10 лабораторных работ по дисциплине «Вычислительные системы, сети и телекоммуникации», изучаемой в базовой части профессионального цикла дисциплин по направлению «Прикладная информатика». При изучении дисциплины используется авторская программа SIM (программная модель ЭВМ). Пособие удовлетворяет требованиям ФГОС. Пособие может также использоваться при изучении таких дисциплин, как «Архитектура компьютера», «Архитектура вычислительных систем», «Программные и аппаратные средства информатики».

## **Введение**

Лабораторные работы, вошедшие в данное пособие, предназначены для изучения устройства ЭВМ с минимальной системой команд, алгоритма функционирования центрального устройства управления (ЦУУ) при выполнении арифметических и посыльных операций, операций передачи управления и ввода-вывода, а также для ознакомления с различными типами адресации: прямой, непосредственной, косвенной, относительной.

Спецификой данного цикла лабораторных работ является то, что изучение алгоритма работы ЦУУ и различных типов адресации осуществляется на примере функционирования простейшей ЭВМ с минимальной системой команд, программная модель которой реализована IBM-совместимой ЭВМ.

Так же студенты получают базовые знания о составе, классификации и функционировании информационных сетей и телекоммуникаций, на практике прорабатываются методы и способы маршрутизации и коммутации каналов и пакетов.

В процессе выполнения лабораторных работ студент формирует и демонстрирует следующие компетенции:

- способен использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатировать современное электронное оборудование и информационно-коммуникационные технологии в соответствии с целями образовательной программы бакалавра (ПК-3);

- способен ставить и решать прикладные задачи в образовании с использованием современных информационно-коммуникационных технологий (ПК-4);

- способен принимать участие в реализации профессиональных коммуникаций в рамках проектных групп, презентовать результаты проектов и обучать пользователей ИС образовательных систем (ПК-14).

# **1. Архитектура ЭВМ с минимальной системой команд**

## ***1.1. Структура ЭВМ***

ЭВМ, программная модель которой позволяет изучать устройство машины, состоит из оперативной памяти (ОП), арифметического устройства (АУ), устройства ввода (УВВ), устройства вывода (УВЫВ) и центрального устройства управления (ЦУУ). Структурная схема ЭВМ показана рис. 1.1.

В ячейках ОП хранятся команды и данные. Емкость ОП – 100 ячеек. По сигналу ЗП выполняется запись содержимого регистра слова (РС) в ячейку с адресом из регистра адреса (РА). По сигналу ЧТ содержимое ячейки памяти с адресом, содержащимся в РА, передается в РС.

АУ по сигналу ПУСК АУ осуществляет выполнение одной из арифметических операций, определяемой кодом операции (КОП), над содержимым аккумулятора (АК) и регистра операнда (РО). Результат операции помещается в АК. При завершении выполнения операции, АУ вырабатывает сигналы признаков результата операции: НУЛЬ=1, если результат равен 0; ЗНАК=1, если результат отрицателен; ПП=1, если при выполнении операции произошло переполнение разрядной сетки. В случаях, когда эти условия не выполняются, соответствующие сигналы имеют нулевое значение.

УВВ и УВЫВ в рассматриваемой ЭВМ, представляют собой регистры РВВ (регистр ввода) и РВЫВ (регистр вывода). Содержимое РВВ может быть передано в АК, содержимое АК – в РВЫВ.

ЦУУ осуществляет выборку команд из ОП в последовательности, определяемой естественным порядком выполнения команд (в порядке возрастания адресов команд) или командами передачи управления; выборку из ОП operandов, задаваемых адресами; инициирование выполнения операции, предписанной командой; останов или переход к выполнению следующей команды.

В состав ЦУУ ЭВМ входят счетчик адреса команды (СЧАК), содержащий адрес команды; регистр команды (РК), содержащий код команды; регистр базового адреса (РБА), содержащий базовый адрес; регистр адреса останова (АО),

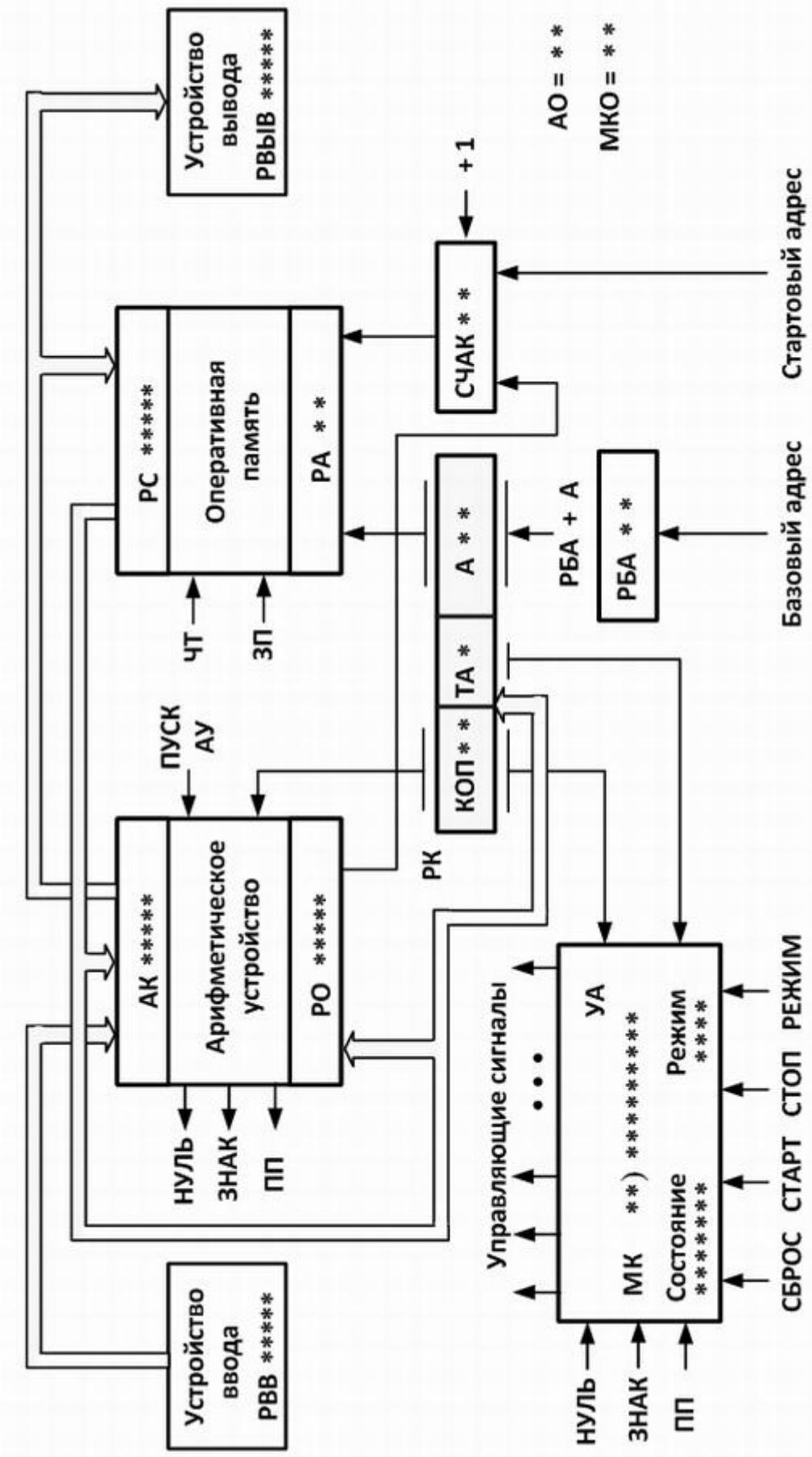


Рис. 1.1. Структурная схема ЭВМ

регистр номера микрокоманды, перед выполнением которой осуществляется останов (МКО); управляющий автомат (УА), вырабатывающий сигналы управления.

Регистры АК, РО, РС, РВВ, РВЫВ, РК имеют длину 5 разрядов, регистры АО, МКО, РБА, РА и СЧАК – 2 разряда. Регистры АО и МКО показаны на структурной схеме условно (рис.1.1).

## 1.2. Представление данных и команд

Информация в ЭВМ – данные и команды представляются словами, содержащими 5 десятичных разрядов. Допустимыми значениями старшего разряда являются 0 и 1. Формат представления данных показан на рис. 1.2.



Рис. 1.2. Формат представления данных

ЭВМ может выполнять операции в десятичной системе счисления над целыми десятичными числами, диапазон изменения которых  $-(10^4 - 1) \div (10^4 - 1)$ . Старший разряд слова используется для кодирования знака. Если результат арифметической операции выходит за пределы этого интервала, что называют переполнением разрядной сетки, то АУ вырабатывает сигнал ПП=1. Результатом операции деления является целая часть частного.

Команды ЭВМ делятся на две группы: команды 1-го типа (безадресные) и команды 2-го типа (адресные). Структура команд 1-го типа показана на (рис . 1.3) и содержит одно поле кода операции – КОП, представленное двумя старшими разрядами. Значение других разрядов при выполнении этих команд не используется, и могут быть произвольными.

В состоянии РАБОТА ЭВМ выполняет программу, представляющую собой последовательность команд. В режиме АВТ переход ЭВМ из состояния РАБОТА в состояние ОСТАНОВ происходит в следующих ситуациях:

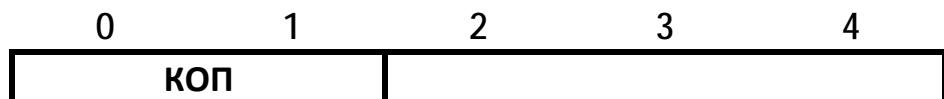


Рис. 1.3. Структура безадресных команд

Структура команд типа 2 показана на рис. 1.4 и содержит три поля: КОП, ТА – поле типа адресации, А – поле адреса.

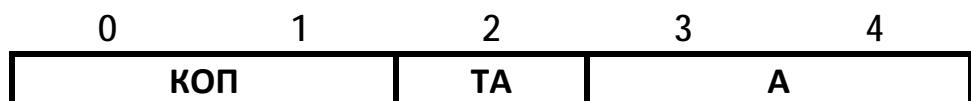


Рис. 1.4. Структура адресных команд

ЭВМ может адресовать операнд, над которым выполняется заданная операция, четырьмя способами в зависимости от значения ТА.

В случае прямой адресации ( $TA=0$ ) исполнительный адрес (ИА), т.е. адрес, по которому в ОП записан операнд, содержится в поле адреса команды.

При непосредственной адресации ( $TA=1$ ) операнд находится в поле адреса команды. Операнды, адресуемые непосредственно, могут принимать значения от 00 до 99.

В случае косвенной адресации ( $TA=2$ ) ИА выбирается из ячейки ОП, адрес которой указан в поле адреса команды.

При относительной адресации ( $TA=3$ ) ИА вычисляется как сумма базового адреса – содержимого РБА и поля адреса команды. Если результат суммирования превышает 99, то в качестве значения ИА берутся два младших разряда суммы.

Значения ТА, правила определения ИА и операнда для различных типов адресации приведены в табл. 1.1.

Таблица 1.1  
Типы адресации

Тип адресации	ТА	ИА	Операнд
Прямая	ПА	0	А
Непосредственная	НА	1	-
Косвенная	КА	2	ОП[А]
Относительная	ОА	3	РБА+А

*Примечание.* ОП [А] – содержимое ячейки ОП с адресом А.

Команды ЭВМ и соответствующие им операции, а также используемые различными командами способы адресации приведены в табл. 1.2 и 1.3. Выделяются 4 группы операций, инициируемых соответствующими командами: операции ввода-вывода (ОВВ), арифметические операции (АОП), посыльные операции (ПО), операции передачи управления (ОПУ).

Таблица 1.2  
Команды типа 1

Команда			Операция	Примечание
КОП				
НОП	00	* * *	-	Пустая операция
ОСТ	12	* * *	ОСТАНОВ	Прекращение работы
Команды ввода-вывода				
ВВ	01	* * *	АК:=РВВ	Ввод
ВЫВ	02	* * *	РВЫК:=АК	Вывод

*Примечание.* Символы «\*» означают, что значения трех младших разрядов кода команды могут быть произвольными.

### **1.3. Состояния и режимы работы ЭВМ**

Ядром ЦУУ являющийся автомат (УА), вырабатывающий сигналы управления, которые инициируют работу АУ и ОП, передачу информации между регистрами устройств ЭВМ и действия над содержимым регистров ЦУУ.

Входными сигналами для УА являются КОП, ТА, НУЛЬ, ЗНАК, ПП, СБРОС, СТАРТ, СТОП и сигналы установки режима работы ЭВМ.

ЭВМ может находиться в одном из двух состояний: ОСТАНОВ, РАБОТА и функционирует в одном из режимов: АВТ, ЦИКЛ, ТАКТ. В состояние РАБОТА ЭВМ переходит по сигналу СТАРТ, в состояние ОСТАНОВ – по сигналу СТОП после завершения выполнения команды. По сигналу СБРОС осуществляется очистка АК, РО, РС, РА, РК, СЧАК; отменяются остановы по адресу команды и по номеру микрокоманды; устанавливается исходное состояние УА.

Таблица 1.3  
Команды типа 2

Команда					Операция	Примечания		
КОП	ТА	А						
Арифметические команды								
СЛ	03	ПА, НА, КА, ОА	0,1,2,3	А	АК:=АК+ОПЕРАНД	Сложение		
ВЫЧ	04	ПА, НА, КА, ОА	0,1,2,3	А	АК:=АК-ОПЕРАНД	Вычитание		
УМН	05	ПА, НА, КА, ОА	0,1,2,3	А	АК:=АК*ОПЕРАНД	Умножение		
ДЕЛ	06	ПА, НА, КА, ОА	0,1,2,3	А	АК:=АК/ОПЕРАНД	Деление		
Посылочные команды								
ЧТ	07	ПА, НА, КА, ОА	0,1,2,3	А	АК:=ОПЕРАНД	Чтение		
ЗП	08	ПА, КА, ОА	0, 2, 3	А	ОП[ИА]:=АК	Запись		
Команды передачи управления								
УП0	09	ПА, КА, ОА	0, 2, 3	А	СЧАК := $\begin{cases} \text{ИА}, & \text{если } \text{АК}=0 \\ \text{СЧАК}+1, & \text{если } \text{АК} \neq 0 \end{cases}$	Условный переход по нулю		
УПЗН	10	ПА, КА, ОА	0, 2, 3	А	СЧАК := $\begin{cases} \text{ИА}, & \text{если } \text{АК}<0 \\ \text{СЧАК}+1, & \text{если } \text{АК} \geq 0 \end{cases}$	Условный переход по знаку		
БП	11	ПА, КА, ОА	0, 2, 3	А	СЧАК:=ИА	Безусловный переход		

*Примечание.* Для ОПУ исполнительный адрес, формируемый в соответствии с табл. 1.1, имеет смысл адреса команды.

В состоянии РАБОТА ЭВМ выполняет программу, представляющую собой последовательность команд. В режиме АВТ переход ЭВМ из состояния РАБОТА в состояние ОСТАНОВ происходит в следующих ситуациях:

- в результате выполнения команд ОСТ;
- перед выполнением команды с адресом, равным заданному адресу останова АО;
- перед выполнением микрокоманды с номером, равным заданному номеру микрокоманды останова МКО;
- при возникновении переполнения разрядной сетки;
- при выполнении команды с некорректным ТА или КОП.

В двух последних случаях останов сопровождается выдачей на терминал сообщения «ПРОГРАМНАЯ ОШИБКА».

В режиме ЦИКЛ переход ЭВМ из состояния РАБОТА в состояние ОСТАНОВ происходит в следующих ситуациях:

- после завершения выполнения очередной команды и выборки следующей;
- перед выполнением микрокоманды с номером, равным заданному номеру микрокоманды останова МКО;
- при возникновении переполнения разрядной сетки;
- при выполнении команды с некорректным ТА или КОП.

В двух последних случаях выборка следующей команды не выполняется и, также как в режиме АВТ, останов сопровождается выдачей на терминал сообщения «ПРОГРАМНАЯ ОШИБКА».

#### **1.4. Взаимодействие пользователя с моделью ЭВМ**

Каждый из пользователей, работая за терминалом, осуществляет взаимодействие с персональной моделью ЭВМ. Номер ЭВМ задается преподавателем. После запуска программы моделирования на экран терминала

выводится текст «SIM> НОМЕР ЭВМ:». Пользователь должен ввести номер ЭВМ и нажать Enter (далее  $\downarrow$ ). Далее программа взаимодействия с пользователем выводит на экран структурную схему ЭВМ.

Если в начале сеанса был введен ошибочный номер ЭВМ или по какой-либо причине в процессе работы изображение структурной схемы ЭВМ на экране терминала было искажено, необходимо ввести директиву НАЧ (от «начало»). При исполнении директивы НАЧ на экран терминала выводится текст «SIM> НОМЕР ЭВМ:». Пользователь может нажать  $\downarrow$  или набрать номер ЭВМ и затем нажать  $\downarrow$ . В первом случае будет продолжена работа с ЭВМ, номер которой был введен ранее, причем состояние модели ЭВМ будет сохранено. Во втором случае будет запрошено подтверждение «СМЕНИТЬ ЭВМ?». В случае ответа «ДА» будет начата работа с новой ЭВМ, при ответе «НЕТ» - продолжена работа с ЭВМ, номер которой был введен ранее. Смена ЭВМ не влечет за собой потери содержимого ОП.

Управление моделью ЭВМ пользователь осуществляет с помощью директив, вводимых им с терминала. Список директив приведен в табл. 1.4. О своей готовности воспринимать очередную директиву программа взаимодействия с пользователем сообщает выводом на экран терминала сообщения «SIM>». Ввод директивы завершается нажатием  $\downarrow$ . До ввода  $\downarrow$  возможно стирание ошибочно введенных символов. Стирание последнего из введенных символов осуществляется нажатием клавиши Backspace. В случае ввода ошибочной директивы на экран терминала выводится текст «ОШИБКА, ПОВТОРИТЕ ВВОД».

С помощью директив пользователь может изменять содержимое регистров ЭВМ, вызывать содержимое ячеек ОП в РС, записывать ОП данные и команды, устанавливать режим работы ЭВМ, запускать, останавливать и осуществлять сброс ЭВМ. При исполнении директив структурная схема ЭВМ остается на экране терминала и изменение содержимого какого-либо регистра, состояния или режима работы ЭВМ приводит к обновлению соответствующей информации на экране терминала.

Заканчивая работу, пользователь может сохранить состояние модели ЭВМ (включая содержимое ОП) в дисковой памяти. Для этого необходимо ввести директиву СОХР. Восстановление состояния модели ЭВМ может быть выполнено на следующем занятии директивой ВОССТ. Последней директивой в сеансе работы с моделью должна быть директива КОН.

Для изменения содержимого какого-либо регистра необходимо ввести имя регистра и нажать  $\downarrow$ . После ввода имени регистра программы взаимодействия с оператором выводит на экран терминала символ «=». Далее надо ввести число – новое содержимое регистра и нажать  $\downarrow$ .

Директивы ЧТ и ЗП осуществляют вызов содержимого ячейки ОП с адресом РА в РС и запись содержимого РС в ячейку ОП с адресом РА. Предусмотрена возможность увеличения или уменьшения на единицу содержимого РА после выполнения операций чтения из ОП и записи в ОП. Операции чтения и записи с инкрементом (увеличением на единицу) РА выполняются директивами ЧТИ и ЗПИ, с декрементом (уменьшением на единицу) РА – директивами ЧТД и ЗПД.

Пользователь имеет возможность повторять выполнение директив СТАРТ, ЧТИ, ЧТД, ЗПИ, ЗПД, СБРОС. Для однократного повторения надо ввести пустую строку, т.е.  $\downarrow$ . Этой возможностью удобно пользоваться при записи программы в ОП, при вызове в РС содержимого подряд расположенных ячеек ОП и при выполнении программы в режимах ЦИКЛ и ТАКТ. В первом случае возникает необходимость многократного повторения директивы ЗПИ или ЗПД, во втором случае – директивы ЧТИ или ЧТД, в третьем – директивы СТАРТ.

**Пример 1.1.** Запись в 10, 11 и 12 ячейки ОП команд 01000, 03010, 12000.

SIM> PA $\downarrow$  = 10 $\downarrow$

SIM> PC $\downarrow$  = 01000 $\downarrow$

SIM> ЗПИ $\downarrow$

SIM> PC $\downarrow$  = 03010 $\downarrow$

SIM> ЗПИ $\downarrow$

SIM> PC $\downarrow$  = 12000 $\downarrow$

SIM> ЗП $\downarrow$

**Пример 1.2.** Вызов в РС содержимого 10, 11 и 12 ячеек ОП.

SIM> PA $\dot{z}$  = 10 $\dot{z}$

SIM> ЧТИ $\dot{z}$

SIM> $\dot{z}$

SIM> $\dot{z}$

*Примечание.* Подчеркнутые символы выводятся на экран терминала программой взаимодействия с пользователем, неподчеркнутые – вводятся пользователем.

По директиве ? на экран терминала выводится список директив управления моделью. После ввода  $\dot{z}$  на экране восстанавливается изображение структурной схемы ЭВМ.

Таблица 1.4  
Директивы управления моделью ЭВМ.

Директива	Выполняемые операции
1	2
АК $\dot{z}$ = * * * * * $\dot{z}$	Установить содержимое АК
РА $\dot{z}$ = * * $\dot{z}$	Установить адрес
РС $\dot{z}$ = * * * * * $\dot{z}$	Установить слово
РБА $\dot{z}$ = * * $\dot{z}$	Установить базовый адрес
РВВ $\dot{z}$ = * * * * * $\dot{z}$	Установить вводимое слово
СЧАК $\dot{z}$ = * * $\dot{z}$	Установит стартовый адрес
АО $\dot{z}$ = * * $\dot{z}$	Установить адрес останова
МКО $\dot{z}$ = * * $\dot{z}$	Установить номер микрокоманды останова
ЧТ $\dot{z}$	PC := ОП[РА]
ЧТИ $\dot{z}$	PC := ОП[РА]; PA := PA + 1
ЧТД $\dot{z}$	PC := ОП[РА]; PA := PA - 1
ЗП $\dot{z}$	ОП[РА] := PC
ЗПИ $\dot{z}$	ОП[РА] := PC; PA := PA + 1
ЗПД $\dot{z}$	ОП[РА] := PC; PA := PA - 1

АВТ	
ЦИКЛ	Установить режим работы ЭВМ
ТАКТ	
СТАРТ	Запустить ЭВМ
СТОП	Остановить ЭВМ
СБРОС	Очистить АК, РО, РС, РК, СЧАК; отменить останов по адресу команды и по номеру микрокоманды; установить исходное состояние УА
?	Повторить последнюю директиву
??	Вызвать подсказку
ТЕСТ	Загрузить ОП в тестовую программу
НАЧ	Восстановить изображение структурной схемы ЭВМ
СОХР	Сохранить состояние модели в дисковой память
ВОССТ	Восстановить состояние модели
КОН	Закончить работу с моделью

## 2. Изучение архитектуры ЭВМ

### 2.1. Лабораторная работа №1. Ознакомление с архитектурой ЭВМ, директивами управления моделью ЭВМ

Для решения с помощью ЭВМ некоторой задачи должна быть разработана программа. Программа на языке ЭВМ представляет собой последовательность команд. Код каждой команды определяет выполняемую операцию, тип адресации и адрес. Выполнение программы, записанной в памяти ЭВМ, осуществляется последовательно по командам в порядке возрастания адресов команд или в порядке, определяемой командами передачи управления.

Для того чтобы получить результат выполнения программы, пользователь должен ввести программу в память ЭВМ, определить если это необходимо, содержимое РВВ и РБА, записать в СЧАК стартовый адрес программы, запустить ЭВМ. Каждое из этих действий выполняется посредством определенной последовательности директив управления моделью ЭВМ.

Приведенные далее примеры показывают возможные последовательности директив управления моделью ЭВМ, обеспечивающие запись в ОП кода команды и выполнение этой команды.

**Пример 2.1.** Запись в ОП и выполнение команды ВВ.

Директива	Комментарии
Запись в ОП команды ВВ	
<u>SIM &gt;PA</u> $\varnothing \equiv 10\varnothing$	PA:=10
<u>SIM &gt;PC</u> $\varnothing \equiv 01000\varnothing$	PC:=01000
<u>SIM &gt;ЗП</u> $\varnothing$	ОП[10]:=01000
Установка режима ЦИКЛ	
<u>SIM &gt;ЦИКЛ</u> $\varnothing$	
Установка вводимой величины	
<u>SIM &gt;РВВ</u> $\varnothing \equiv -20\varnothing$	РВВ:=10020
Установка стартового адреса	

<u>SIM &gt;СЧАК</u> $\dot{z} \equiv 10$ $\dot{z}$	СЧАК:=10
Выполнение команды ВВ	
<u>SIM &gt;СТАРТ</u> $\dot{z}$	Выборка команды; РК:=01000
<u>SIM &gt;</u> $\dot{z}$	Выполнение операции; АК:=10020

**Пример 2.2.** Запись в ОП и выполнение команды СЛ с непосредственной адресацией.

Директива	Комментарий
Запись в ОП команды СЛ	
<u>SIM &gt;РА</u> $\dot{z} \equiv 11$ $\dot{z}$	РА:=11
<u>SIM &gt;PC</u> $\dot{z} \equiv 03130$ $\dot{z}$	РС:=03130
<u>SIM &gt;ЗП</u> $\dot{z}$	ОП[11]:=03130
Установка первого операнда в АК	
<u>SIM &gt;АК</u> $\dot{z} \equiv -792$ $\dot{z}$	АК:=10792
Установка стартового адреса	
<u>SIM &gt;СЧАК</u> $\dot{z} \equiv 11$ $\dot{z}$	СЧАК:= 11
Выполнение команды СЛ	
<u>SIM &gt;СТАРТ</u> $\dot{z}$	Выборка команды; РК:=03130 (предполагается, что установлен режим ЦИКЛ)
<u>SIM &gt;</u> $\dot{z}$	Выполнение операции; АК:=10762

**Пример 2.3.** Запись в ОП и выполнение команды ЧТ с косвенной адресацией.

Директива	Комментарий
Запись ОП команды ЧТ	
<u>SIM &gt;РА</u> $\dot{z} \equiv 12$ $\dot{z}$	РА:= 12
<u>SIM &gt;PC</u> $\dot{z} \equiv 07215$ $\dot{z}$	РС:= 07215
<u>SIM &gt;ЗП</u> $\dot{z}$	ОП[12]:= 07215
Запись в ОП исполнительного адреса	
<u>SIM &gt;РА</u> $\dot{z} \equiv 15$ $\dot{z}$	РА:=15
<u>SIM &gt;PC</u> $\dot{z} \equiv 20$ $\dot{z}$	РС:= 00020
<u>SIM &gt;ЗП</u> $\dot{z}$	ОП[15]:= 00020
Запись в ОП операнда	

<u>SIM &gt;PA</u> $\dot{z}$ <u>=20</u> $\dot{z}$	PA:= 20
<u>SIM &gt;PC</u> $\dot{z}$ <u>= - 5</u> $\dot{z}$	PC:= 10005
<u>SIM &gt;ЗП</u> $\dot{z}$	ОП[20]:=10005
Установка стартового адреса	
<u>SIM &gt;СЧАК</u> $\dot{z}$ <u>= 12</u> $\dot{z}$	СЧАК:=12
Выполнение команды ЧТ	
<u>SIM &gt;СТАРТ</u> $\dot{z}$	Выборка команды; РК:= 07215 (предполагается, что установлен режим ЦИКЛ)
<u>SIM &gt;</u> $\dot{z}$	Выполнение операций; АК:= 10005

**Пример 2.4.** Запись в ОП и выполнение команды УПЗН с относительной адресацией при отрицательном и положительном значении содержимого АК.

Директивы	Комментарии
Запись в ОП команды УПЗН	
<u>SIM &gt;PA</u> $\dot{z}$ <u>=13</u> $\dot{z}$	PA =13
<u>SIM &gt;PC</u> $\dot{z}$ <u>=10300</u> $\dot{z}$	PC =10300
<u>SIM &gt;ЗП</u> $\dot{z}$	ОП (13):=10300
Установка в РБА базового адреса	
<u>SIM &gt;РБА</u> $\dot{z}$ <u>=10</u> $\dot{z}$	РБА:=10
Установка содержимого АК	
<u>SIM &gt;АК</u> $\dot{z}$ <u>= - 20</u> $\dot{z}$	АК:= 10020
Установка стартового адреса	
<u>SIM &gt;СЧАК</u> $\dot{z}$ <u>=13</u> $\dot{z}$	СЧАК:= 13
Выполнение команды УПЗН	
<u>SIM &gt;СТАРТ</u> $\dot{z}$	Выборка команды; РК:=10300 (предполагается, что установлен режим ЦИКЛ)
<u>SIM &gt;</u> $\dot{z}$	Выполнение операции; СЧАК:= 10
Установка содержимого АК	
<u>SIM &gt;АК</u> $\dot{z}$ <u>=20</u> $\dot{z}$	АК:= 00020
Установка стартового адреса	
<u>SIM &gt;СЧАК</u> $\dot{z}$ <u>=13</u> $\dot{z}$	СЧАК:= 13
Выполнение команды УПЗН	
<u>SIM &gt;СТАРТ</u> $\dot{z}$	Выборка команды; РК:= 10300
<u>SIM &gt;</u> $\dot{z}$	Выполнение операции; СЧАК:= 14

Пример 2.4 иллюстрирует возможные результаты выполнения команды УПЗН в зависимости от знака содержимого АК. Аналогичным образом выполняется команда УПО. Эти команды обеспечивают возможность разветвления вычислительного процесса в зависимости от результата выполнения предыдущей команды.

Действия пользователя при записи в ОП программы и её выполнении в режиме ЦИКЛ аналогичны рассмотренным в примерах 2.1 – 2.4. Программа обычно располагается в ОП подряд, в порядке возрастания адреса, поэтому при записи программы в ОП удобно пользоваться директивой ЗПИ, при проверке содержимого памяти – директивой ЧТИ (см. примеры 1.1, 1.2). Из примеров 2.1 – 2.4 видно, что для выполнения команды надо два раза ввести директиву СТАРТ. При выполнении последовательности команд необходимость в этом отпадает, так как по первой директиве СТАРТ осуществляется выборка первой команды, по каждой следующей директиве СТАРТ – подготовка адреса следующей команды, формирование исполнительного адреса для текущей команды, выполнение операции, выборка следующей команды.

## ЗАДАНИЕ

1. Ознакомиться с архитектурой ЭВМ (раздел 1 настоящих указаний).
2. Выполнить четыре команды, записывая их в ОП подряд, в порядке этом записывать в таблицу (см. приложение 1) исходное содержимое регистров ЭВМ и их содержимое после выполнения каждой директивы. В каждую строку таблицы записываются текущее состояние регистров ЭВМ и вводимая директива, состояние регистров после выполнения этой директивы записывается в следующую строку таблицы. Варианты заданий приведены в табл. 2.1. Адреса для размещения команд и данные выбрать самостоятельно.
3. Проверить, пользуясь директивой содержимого ЧТИ, содержимое ячеек ОП, в которых записаны команды.
4. Выполнить в режиме ЦИКЛ последовательность из четырех команд.

5. Если командой 4 является команда условной передачи управления, то изменить один из операндов таким образом, чтобы выполнялся альтернативный вариант передачи управления, и повторить выполнение в режиме ЦИКЛ последовательности из 4 команд.

Таблица 2.1

Варианты заданий

№ варианта	Команда 1		Команда 2		Команда 3		Команда 4	
	КОП	ТА	КОП	ТА	КОП	ТА	КОП	ТА
1	ВВ		СЛ	ОА	ЧТ	КА	УП0	ПА
2	ВВ		ВЫЧ	КА	ЗП	ОА	УПЗН	ПА
3	ВВ		УМН	НА	ЗП	ПА	БП	КА
4	ВВ		ДЕЛ	ПА	ЧТ	ОА	УПЗН	КА
5	ВВ		ВЫЧ	НА	ЧТ	ПА	БП	ОА
6	ВВ		УМН	ПА	ЗП	КА	УП0	ОА
7	ВВ		ДЕЛ	ОА	ЗП	КА	БП	ПА
8	ВВ		СЛ	КА	ЧТ	ПА	УПЗН	ПА
9	ВВ		УМН	КА	ЧТ	НА	УПЗН	ОА
10	ВВ		ДЕЛ	НА	ЗП	ОА	УП0	КА
11	ВВ		СЛ	ПА	ЗП	КА	БП	ОА
12	ВВ		ВЫЧ	ОА	ЧТ	НА	УП0	ПА

Содержание отчета

1. Формулировка варианта задания (см. п.2 «Задания»).
2. Результаты выполнения директив управления моделью ЭВМ (см. приложение 1) при записи в ОП и выполнении каждой команды.
3. Последовательность директив, обеспечивающая проверку содержимого ячеек ОП, в которых записаны команды.
4. Директивы, обеспечивающие выполнение в режиме ЦИКЛ последовательности из четырёх команд.

## **2.2. Лабораторная работа №2. Программирование алгоритма без цикла**

Последовательность этапов разработки программы на языке ЭВМ упрощенно можно представить следующим образом: разработка алгоритма решения задачи, разработка программы с символическими адресами, распределение памяти для программы и данных, получение программы с действительными адресами, ввод программы в ОП, отладка программы.

Обычно область размещения данных в ОП находится после программы, возможен вариант расположения данных перед программой. В первом случае для определения начального адреса области данных необходимо знать длину программы, во втором – начальный адрес программы может быть определен, если известно количество ячеек ОП, требуемое для размещения данных. Это обстоятельство обуславливает необходимость разработки программы с символическими адресами. Кроме того, в процессе отладки программы может возникнуть необходимость включения в неё дополнительных команд или размещения дополнительных данных. В этом случае изменения удобно вносить в программу с символическими адресами. Однако распределение памяти и получение программы с действительными адресами придется выполнять заново.

Использование относительной адресации существенно упрощает программирование, так как дает возможность разрабатывать программу с действительными адресами и распределять память для данных, предполагая, что программа и данные размещаются в ОП, начиная с одного и того же адреса. (обычно 00). Необходимость разработки программы с символическими адресами в этом случае отпадает. Понятно, что во время выполнения программы данные и команды программы не могут занимать одну и ту же область памяти. Существуют два пути решения этого вопроса: создание перемещаемой области данных, разработка перемещаемой программы.

Для создания перемещаемой области данных необходимо во всех арифметических и посыльных командах программы вместо прямой адресации использовать относительную. Это обеспечивает возможность смещения области расположения данных путем задания базового адреса. Следует отметить, что такой путь разработки программ для рассматриваемой ЭВМ возможен в случае отсутствия необходимости использования косвенной адресации в арифметических и посыльных командах.

Для получения перемещаемой программы необходимо использовать относительную адресацию в командах передачи управления. Это обеспечивает возможность записывать программу в ОП, начиная с любого адреса. Настройка программы по месту расположения в памяти выполняется заданием базового адреса. Такой подход оставляет возможность использования косвенной адресации в арифметических и посыльных командах.

Приведенные далее примеры иллюстрируют разработку программы без использования относительной адресации и с её использованием для создания перемещаемой области данных.

**Пример 2.5.** Разработка программы для вычисления значения функции

$$y = \begin{cases} (x - 5)^2, & x \geq 5 \\ x^2 + 10x, & x < 5 \end{cases}$$

Граф-схема алгоритма решения задачи показана на рис 2.1.

Заметим, что допустимый диапазон представления чисел в рассматриваемой ЭВМ ограничивает при вычислении заданной функции возможные значения аргумента снизу и сверху:  $-99 \leq x \leq 104$ .

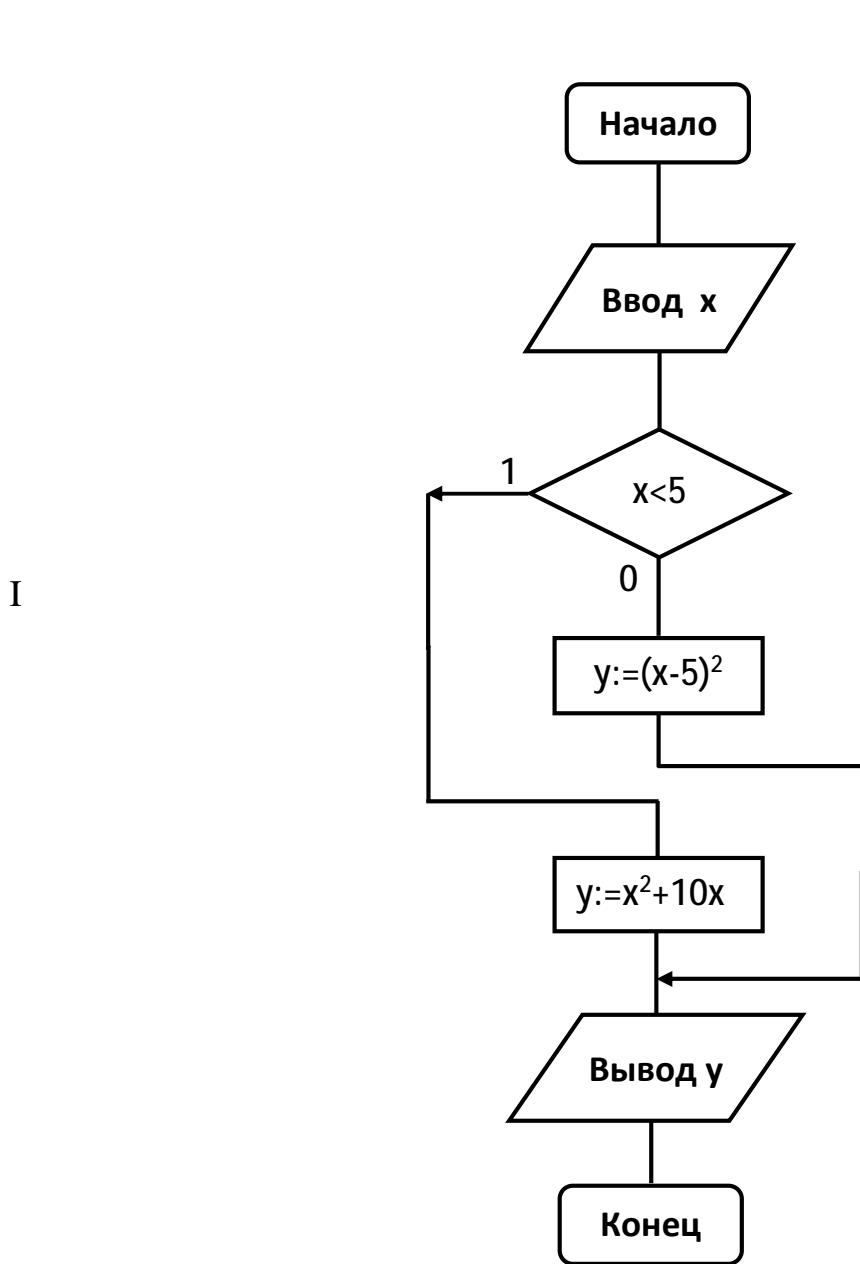


Рис. 2.1. Граф-схема решения задачи

Таблица 2.2

## Программа с символическими адресами

Адрес	Символическая команда			Комментарий
	КОП	ТА	А	
A	ВВ	-	-	AK:=x
A+1	ЗП	П	X	ОП[X]:=x
A+2	ВЫЧ	Н	05	AK:=x-5
A+3	УПЗН	П	A+7	Перейти по адресу A+7, если x-5<0 (x<5)
A+4	ЗП	П	R	ОП [R]:= x-5
A+5	УМН	П	R	AK:=(x-5) <sup>2</sup>
A+6	БП	П	A+13	Перейти по адресу A+13
A+7	ЧТ	П	X	AK:=x
A+8	УМН	Н	10	AK:=10x
A+9	ЗП	П	R	ОП [R]:= 10x
A+10	ЧТ	П	X	AK:=x
A+11	УМН	П	X	AK:=x <sup>2</sup>
A+12	СЛ	П	R	AK:=x <sup>2</sup> +1 0x
A+13	ВЫВ	-	-	РВЫВ:=y
A+14	ОСТ	-	-	ОСТАНОВ
X				x
R				x-5, 10x

Распределение памяти: A= 00

X=15

R=16

Таблица 2.3

## Программа с действительными адресами:

Адрес	Символическая команда			Код команды	Комментарий
	КОП	ТА	А		
00	ВВ	-	-	01000	AK:=x
01	ЗП	П	15	08015	ОП [15]:= x
02	ВЫЧ	Н	05	04105	AK:=x-5

03	УПЗН	П	07	10007	Перейти по адресу 07, если $x-5 < 0$ ( $x < 5$ )
04	ЗП	П	16	08016	ОП [16]:=x-5
05	УМН	П	16	05016	АК:=(x-5) <sup>2</sup>
06	БП	П	13	11013	Перейти по адресу 13
07	ЧТ	П	15	07015	АК:=x
08	УМН	Н	10	05110	АК:=10x
09	ЗП	П	16	08016	ОП [16]:=10x
10	ЧТ	П	15	07015	АК:=x
11	УМН	П	15	05015	АК:=x <sup>2</sup>
12	СЛ	П	16	03016	АК:=x <sup>2</sup> +10x
13	ВЫВ	-	-	02000	РВЫВ:=y
14	ОСТ	-	-	12000	ОСТАНОВ
15					x
16					x-5, 10x

**Пример 2.6.** Программирование алгоритма вычисления значения функции (см.рис.2.1) с использованием относительной адресации для создания перемещаемой области данных.

Таблица 2.4  
Программа с относительными и непосредственными адресами операндов:

Адрес	Символическая команда			Код команды	Комментарий
	КОП	ТА	А		
00	ВВ	-	-	01000	АК:=x
01	ЗП	О	00	08300	ОП[РБА+00]:=x
02	ВЫЧ	Н	05	04150	АК:=x-5
03	УПЗН	П	07	10007	Перейти по адресу 07, если $x-5 < 0$ ( $x < 5$ )
04	ЗП	О	01	08301	ОП [РБА+01]:=x-5
05	УМН	О	01	05301	АК:=(x-5) <sup>2</sup>
06	БП	П	13	11013	Перейти по адресу 13
07	ЧТ	О	00	07300	АК:=x
08	УМН	Н	10	05110	АК:=10x
09	ЗП	О	01	08301	ОП[РБА+01]:=10x
10	ЧТ	О	00	07300	АК:=x
11	УМН	О	00	05300	АК:=x <sup>2</sup>
12	СЛ	О	01	03301	АК:=x <sup>2</sup> +10x

13	ВЫВ	-	-	02000	РВЫВ:=у
14	ОСТ	-	-	12000	ОСТАНОВ

Размещение данных в ОП:

Относительный адрес	Данные
00	x
01	x-5, 10x

Для того, чтобы область данных располагалась в ОП, после программы, базовый адрес должен быть равен 15.

Отладка программы – это процесс нахождения и устранения в ней ошибок. Осуществляя проверку результатов выполнения программы в режиме АВТ, необходимо убедиться в том, что:

- 1) Остановка происходит по команде ОСТ, предусмотренной в программе;
- 2) Верны результаты вычислений для различных наборов исходных данных, обеспечивающих выполнение программы в соответствии с возможными путями граф-схемы алгоритма.

Если выполнение программы в режиме АВТ не приводит к получению верных результатов, то, прослеживая выполнение программы в режиме ЦИКЛ, необходимо найти команду, являющуюся причиной ошибки и исправить её.

## ЗАДАНИЕ

1. Разработать программу вычисления и вывода значения функции:

$$y = \begin{cases} f_i(x), & x \geq a \\ f_j(x), & x < a \end{cases}$$

для вводимого значения аргумента  $x$ . Использовать в программе относительную адресацию для создания перемещаемой области данных. Функции и допустимые пределы изменения аргумента приведены в табл. 2.5, варианты заданий – в табл. 2.6.

2. Записать программу в ОП ЭВМ, для этого:

- 2.1. Записать в РА адрес первой команды программы.
- 2.2. Записать в РС команду, которая должна размещаться в ОП по адресу , содержащемуся в РА.
- 2.3. Внести директиву ЗПИ.
- 2.4. Повторять выполнение п.п. 2.2, 2.3 до тех пор, пока в ОП не будет записана вся программа.
3. Отладить программу, для этого:
  - 3.1. Записать в РВВ значение аргумента  $x^3 a$ .
  - 3.2. Записать в РБА базовый адрес.
  - 3.3. Установить режим АВТ.
  - 3.4. Записать в СЧАК стартовый адрес программы.
  - 3.5. Проверить правильность выполнения программы в режиме АВТ (правильность результата и адреса останова), в случае наличия ошибки выполнить п.п. 3.6-3.8.
  - 3.6. Установить режим ЦИКЛ.
  - 3.7. Записать в СЧАК стартовый адрес программы.
  - 3.8. Наблюдая выполнение программы в режиме ЦИКЛ, найти команду, являющуюся причиной ошибки; исправит её; выполнить п.п. 3.3-3.5.
  - 3.9. Записать в РВВ значение аргумента  $x < a$ ; выполнить п.п. 3.3-3.5.
4. Для выбранного значения аргумента  $x$ , наблюдая выполнение отлаженной программы в режиме ЦИКЛ, записать в таблицу (см. приложение 2) содержимое регистров ЭВМ перед выполнением каждой команды.

Таблица 2.5

Функции и допустимые пределы изменения аргумента

$k$	$f_k(x)$	$x_{min}$	$x_{max}$	$k$	$f_k(x)$	$x_{min}$	$x_{max}$
1	$\frac{x+17}{1-x}$	2	12	5	$\frac{(x+2)^2}{15}$	50	75
2	$\frac{(x+3)^2}{x}$	1	50	6	$\frac{2x^2+7}{x}$	1	30

3	$\frac{1000}{x+10}$	-50	-15	7	$\frac{x^2 - 2x}{10}$	-50	50
4	$(x - 1)^3$	-20	20	8	$\frac{8100}{x^2}$	1	90

Таблица 2.6

## Варианты задания

№ варианта	$i$	$j$	$a$	№ варианта	$i$	$j$	$a$
1	2	1	12	7	8	6	30
2	4	3	-20	8	2	6	25
3	8	4	20	9	5	7	50
4	6	1	12	10	2	4	18
5	5	2	50	11	8	1	12
6	7	3	15	12	7	6	25

## Содержание отчета

1. Формулировка варианта задания (см. п.1 «Задания», табл. 2.5, 2.6).
2. Граф-схема алгоритма решения задачи.
3. Программа.
4. Размещение данных в ОП, значение базового адреса.
5. Последовательность состояний регистров ЭВМ при выполнении программы в режиме ЦИКЛ (см. приложение 2) для одного значения аргумента.
6. Результаты выполнения программы для пяти значений аргумента, выбранных самостоятельно.

## **2.3. Лабораторная работа №3. Программирование цикла с переадресацией**

При решении задач, связанных с обработкой массивов, возникает необходимость изменения исполнительного адреса при повторном выполнении некоторых команд. Эта задача может быть решена путем использования косвенной адресации и модификации содержимого ячейки ОП с косвенным адресом.

Необходимость косвенной адресации в арифметических и посыльных командах исключает возможность использования в программе относительной адресации для создания перемещаемой области данных. В таких случаях имеет смысл разрабатывать перемещаемую программу.

Для получения перемещаемой программы необходимо использовать относительную адресацию в командах передачи управления. Это обеспечит возможность расположения программы в произвольной области ОП. Настройка программы по месту расположения осуществляется заданием базового адреса. Приведенный далее пример иллюстрирует разработку перемещаемой программы.

**Пример 2.7.** Разработка программы вычисления суммы чисел  $C_1, C_2, \dots, C_N$ .

Граф-схема алгоритма решения задачи показана на рис. 2.2.

Исходными данными в этой задаче являются  $N$  – количество суммируемых чисел и  $C_1, C_2, \dots, C_N$  – суммируемые числа. Заметим, что должно выполняться условие  $N \geq 1$ , т.к. алгоритм предусматривает по крайней мере одно суммирование. Кроме того, предполагается, что суммируемые числа записаны в ОП подряд. Результатом является сумма  $S$ . Используемые для решения задачи промежуточные переменные имеют следующий смысл:  $A_1$  – адрес  $C_1$ ,  $A_i$  – адрес  $C_i$ ,  $A_{N+1}$  – увеличенный на единицу адрес  $C_N$ .

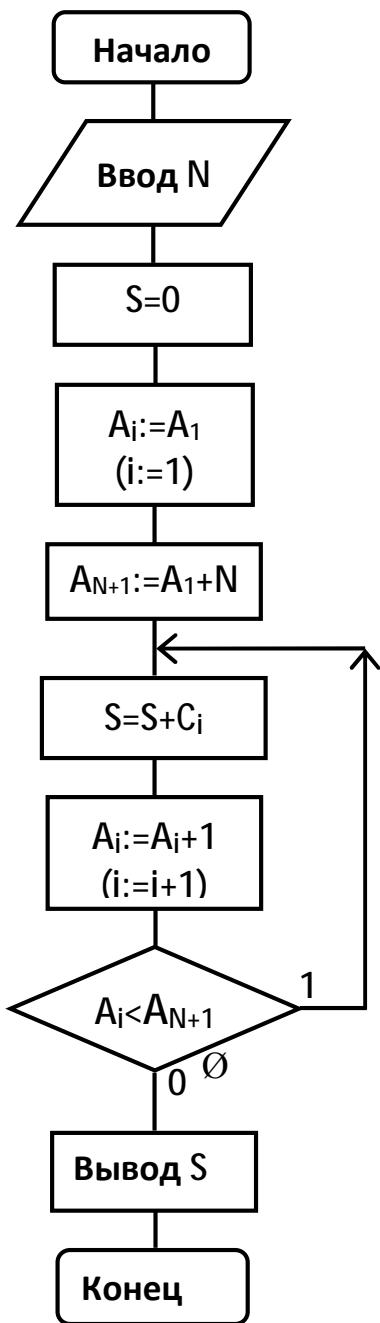


Рис. 2.2. Граф-схема решения задачи

Таблица 2.7

Программа с действительными адресами операндов и относительными адресами переходов

Относительный адрес	Символическая команда			Код команды	Комментарий
	КОП	ТА	А		
00	ВВ	-	-	01000	АК:=N
01	ЗП	П	00	08000	ОП [00]:=N
02	ЧТ	Н	00	07100	АК:= 0
03	ЗП	П	01	08001	S:= 0
04	ЧТ	П	02	07002	АК:=A <sub>1</sub>
05	ЗП	П	03	08003	A <sub>i</sub> :=A <sub>1</sub>
06	СЛ	П	00	03000	АК:=A <sub>1</sub> + N
07	ЗП	П	04	08004	A <sub>N+1</sub> :=A <sub>1</sub> +N
08	ЧТ	П	01	07001	АК:=S
09	СЛ	К	03	03203	АК:=S+ОП [A <sub>i</sub> ]
10	ЗП	П	01	08001	S :=S+ОП [A <sub>i</sub> ]
11	ЧТ	П	03	07003	АК:=A <sub>i</sub>
12	СЛ	Н	01	03101	АК:=A <sub>i</sub> +1
13	ЗП	П	03	08003	A <sub>i</sub> := A <sub>i</sub> +1
14	ВЫЧ	П	04	04004	АК:=A <sub>i</sub> -A <sub>N+1</sub>
15	УПЗН	О	08	10308	Перейти по адресу РБА+08, если A <sub>i</sub> <A <sub>N+1</sub>
16	ЧТ	П	01	07001	АК:=S
17	ВЫВ	-	-	02000	РВЫВ:=S
18	ОСТ	-	-	12000	ОСТАНОВ

В процессе разработки программы для переменных N, S, A<sub>1</sub>, A<sub>i</sub>, A<sub>N+1</sub> выделены ячейки ОП с адресами соответственно 00, 01, 02, 03, 04. Программа может быть размещена в ОП начиная с ячейки 05. Базовый адрес в этом случае должен быть равен 05. Суммируемые числа могут быть записаны в ОП начиная с ячейки 24. В этом случае в ячейку 02 должна быть записана константа 00024.

## ЗАДАНИЕ

1. Разработать перемещаемую программу определения заданной характеристики последовательности чисел  $C_1, C_2, \dots, C_N$ . Варианты заданий приведены в табл. 2.8.
2. Записать программу, необходимые константы и исходные данные в ОП ЭВМ.
3. Отладить программу.

Таблица 2.8

### Варианты задания

№ Варианта	Характеристика последовательности чисел $C_1, C_2, \dots, C_N$
1	Количество положительных чисел
2	Номер минимального числа
3	Произведение всех чисел
4	Номер первого отрицательного числа
5	Количество чисел, равных $C_1$
6	Количество отрицательных чисел
7	Максимальное число
8	Номер первого положительного числа
9	Минимальное число
10	Номер максимального числа
11	Количество нулей
12	Количество чисел, меньших $C_1$

### Содержание отчета

1. Формулировка варианта задания (см. п.1 «Задания», таб. 2.8).
2. Граф-схема алгоритма решения задачи.
3. Программа.
4. Размещение в ОП переменных, программы и необходимых констант, значение базового адреса.
5. Значения исходных данных и результата выполнения программы.

### **3. Изучение алгоритма работы центрального устройства управления ЭВМ**

#### ***3.1. Принцип микропрограммного управления***

Каждая команда, выполняемая ЭВМ, может быть представлена в виде последовательности элементарных действий над словами информации. Эти действия называются микрооперациями. Для управления порядком следования микроопераций используются логические условия, которые в зависимости от значения слов, преобразуемых микрооперациями, принимают значения 1 или 0. Таким образом, выполнение некоторой микрооперации может привести к изменению значений определенных логических условий и, следовательно, оказать влияние на порядок выполнения других микроопераций.

Микрооперации выполняются под воздействием управляющих сигналов, вырабатываемых УА в определенной последовательности в соответствии с алгоритмом функционирования ЦУУ. Алгоритм работы ЦУУ, представленный в терминах микроопераций и логических условий, называется микропрограммой. Микропрограмма хранится в памяти УА в виде последовательности микрокоманд (МК) и определяет порядок выполнения микроопераций и проверки логических условий.

Выполнение каждой микрооперации заключается в присваивании некоторому слову или части слова значения выражения. По форме записи, как правило, микрооперация – это оператор присваивания. При записи таких микроопераций слева от знака присваивания указывается имя регистра или поля регистра, куда пересыпается значение выражения. Справа от знака присваивания записывается выражение, содержащее имена регистров, полей регистров и константы, соединенные знаками операций. Частными случаями выражения являются имя регистра, имя поля регистра, константа. Операция присваивания понимается следующим образом. Если разрядность регистра или поля регистра, имя которого записано в левой части, равна  $l$ , а выражение в правой части

содержит  $r$  разрядов, то при  $l \leq r$  слову присваивается  $l$  младших разрядов выражения, а при  $l > r$  старшим ( $l - r$ ) разрядам присваивается значения 0 и  $r$  младшим разрядам – значение выражения.

В соответствии с видом выражения выделяют следующие группы микроопераций: установки, передачи, счёта, сложения. Микрооперация установки – это присваивание слову или части слова значения константы. Правой частью в записи микрооперации установки является константа. Микрооперация передачи – это присваивание слову или части слова значения другого слова или его части. В правой части в этом случае записывается имя регистра или поля регистра. Микрооперация счёта обеспечивает изменение значения слова на единицу, микрооперация сложения – присваивание слову значения суммы слагаемых.

### **Пример 3.1.** Примеры микроопераций

АК:= 0	– микрооперация установки
АК:= РВВ	– микрооперация передачи
СЧАК:= СЧАК + 1	– микрооперация счёта
А:= РБА + А	– микрооперация сложения

Логическое условие – это булево выражение, состоящее из первичных булевых выражений, связанных знаками операций булевой алгебры: отрицания, конъюнкции, дизъюнкции. Первичными булевыми выражениями являются одноразрядные двоичные слова и отношения. Отношения имеют вид  $A^*B$ , где  $A, B$  – некоторые слова либо константы,  $*$  – знак операции отношения  $=, ^1, \neq, <, >$ .

### **Пример 3.3.** Примеры логических условий

СТАРТ – логическое условие совпадает по значению со словом СТАРТ.

СЧАК = АО – логическое условие равно 1, если равенство выполняется, и равно 0 в противном случае.

### **3.2. Микропрограмма центрального устройства управления ЭВМ**

Выполнение каждой команды в ЭВМ происходит в соответствии с микропрограммой ЦУУ. Управляющие сигналы, вырабатывающие УА, вызывают выполнение микроопераций в последовательности, определяемой значениями логических условий.

Граф-схема микропрограммы ЦУУ показана на рис. 3.1. Условные вершины, изображаемые ромбами, обозначают анализ значений логических условий. Операторные вершины, изображаемые прямоугольниками, обозначают выполнение одной или нескольких микроопераций. Данная граф-схема представляет укрупненный алгоритм функционирования ЦУУ, так как в этом алгоритме выделены, но не раскрыты детально, такие действия как выборка команды, формирования исполнительного адреса (ИА), выполнение операций ввода-вывода (ОВВ), арифметических операций (АОП), посыльных операций (ПО), операций передачи управления (ОПУ). Каждое из этих действий может быть представлено в виде отдельного алгоритма, детализация тих действий приведена на рис. 3.2 – 3.7.

В общем случае, процесс выполнения команды (основной машинный цикл) может быть представлен следующей последовательностью этапов ( см. рис. 3.1.):

1. Выборка команды;
2. Подготовка адреса следующей команды;
3. Формирование исполнительного адреса;
4. Выполнение операции.

Под выборкой команды понимается чтение кода команды из ОП и передача его в РК.

Подготовка адреса следующей команды заключается в увеличении на I содержимого счетчика адреса команды (СЧАК), что соответствует естественному порядку следованию команд.

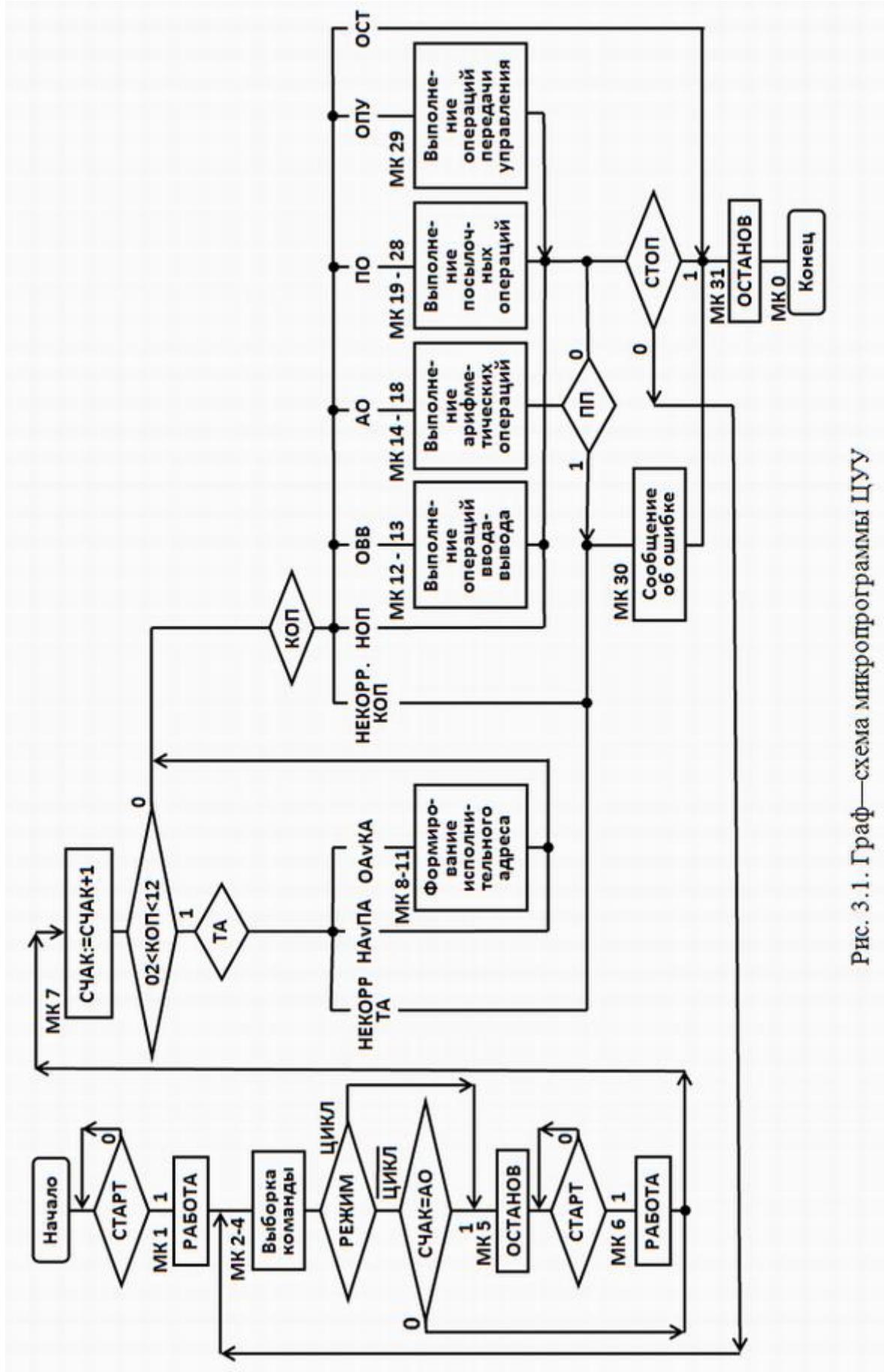


Рис. 3.1. Граф—схема микропрограммы ЦУУ

Формирование исполнительного адреса заключается в определении исполнительного адреса ОП (для арифметических и посыльных операций) либо исполнительного адреса следующей команды (для операций передачи управления).

При выполнении операции осуществляются действия, предписанные КОП. Каждая из групп операций: ОВВ, АОП, ПО, ОПУ – выполняется в соответствии с определенным алгоритмом.

*Примечание.* В циклах команд типа 1 исполнительный адрес не формируется

Рассмотрим микрооперации и логические условия, необходимые для понимания общего алгоритма ЦУУ (см. рис. 3.1). Микрооперации РАБОТА, ОСТАНОВ, СООБЩЕНИЕ ОБ ОШИБКЕ относятся к группе микроопераций установки. РАБОТА и ОСТАНОВ – выполняют установку соответствующего состояния ЭВМ, СООБЩЕНИЕ ОБ ОШИБКЕ заключается в установке признака останова по ошибке, что в реализованной модели ЭВМ приводит к выдаче на терминал пользователя текстового сообщения «ПРОГРАММНАЯ ОШИБКА». Микрооперация СЧАК:=СЧАК+1 является микрооперацией счета и обеспечивает подготовку адреса следующей команды.

Логические условия СТАРТ, СТОП, ПП принимают значение 1 при единичных значения соответствующих сигналов. Условие СТАРТ проверяется перед выполнением микрооперации РАБОТА и, если сигнал СТАРТ = 1, то ЭВМ переключается в состояние РАБОТА. Условие СТОП проверяется после выполнения операции и, если сигнал СТОП = 1, то ЭВМ переключается в режим ОСТАНОВ, если СТОП = 0, то основной машинный цикл повторяется и начинается выборка следующей команды. Условие ПП проверяется после выполнения АОП и, если АУ вырабатывает сигнал ПП = 1, то на экран терминала выдается сообщение об ошибке и ЭВМ переключается в состояние ОСТАНОВ.

Значения логических условий РЕЖИМ = ЦИКЛ и РЕЖИМ = АВТ ∨ ТАКТ определяются режимом работы ЭВМ и проверяются после выборки команды. В режиме ЦИКЛ по завершении выборки команды ЭВМ переключается в состояние ОСТАНОВ и работа может быть продолжена по сигналу СТАРТ = 1.

Значение логического условия СЧАК = АО определяется содержимым СЧАК (АО не изменяется в процессе выполнения программы) и может быть равно 1, если задан останов по адресу команды. Останов по адресу команды происходит после выборки команды, адрес которой равен АО. После останова работа может быть продолжена по сигналу СТАРТ = 1.

Значения 0-го и 1-го разрядов РК определяют логические условия  $02 < \text{КОП} < I2$ ,  $\text{КОП} = \text{НЕКОРР}$ .  $\text{КОП}$ ,  $\text{КОП} = \text{НОП}$ ,  $\text{КОП} = \text{ОВВ}$ ,  $\text{КОП} = \text{АОП}$ ,  $\text{КОП} = \text{ПО}$ ,  $\text{КОП} = \text{ОПУ}$ ,  $\text{КОП} = \text{ОСТ}$ . Условие  $02 < \text{КОП} < I2$  проверяется с целью определения типа выполняемой команды. Для команд типа 1 (безадресных команд) формирование ИА не осуществляется, после выборки команды и подготовки адреса следующей команды начинается выполнение операции. Если  $\text{КОП} = \text{НЕКОРР}$ .  $\text{КОП}$ , что соответствует условию  $\text{КОП} > I2$ , на экран терминала выдается сообщение о программной ошибке и ЭВМ переключается в состояние ОСТАНОВ. Проверка остальных логических условий, связанных с КОП, позволяет выявить группу операций или отдельную операцию, подлежащих выполнению.

Значение 2-го разряда РК определяет логические условия  $\text{ТА} = \text{НЕКОРР}$ .  $\text{ТА}$ ,  $\text{ТА} = \text{НА} \vee \text{ПА}$ ,  $\text{ТА} = \text{ОА} \vee \text{КА}$ . Если  $\text{ТА} = \text{НЕКОРР}$ .  $\text{ТА}$ , что соответствует условию  $\text{ТА} > 3$ , на экран терминала выдается сообщение о программной ошибке и ЭВМ переключается в состояние ОСТАНОВ. Условие  $\text{ТА} = \text{ОА} \vee \text{КА}$  проверяется с целью определения необходимости формирования ИА, которое при выполнении команд типа 2 (адресных команд) осуществляется только в случае относительной или косвенной адресации. При выполнении условия  $\text{ТА} = \text{НА} \vee \text{ПА}$  исполнительный адрес не формируется, так как при прямой адресации после выборки команды содержимое поля адреса РК уже является исполнительным адресом, а при непосредственной адресации понятие ИА не существует, поскольку в поле адреса РК записан операнд.

Номер микрооперации останова микропрограммы записывается в регистр МКО (см. рис. 1.1). Это значит, что после выполнения микрооперации с номером, равным МКО, работа ЦУУ может быть продолжена сигналу СТАРТ = 1.

### ***3.3. Работа модели ЭВМ в режиме ТАКТ***

Режим ТАКТ обеспечивает возможность изучения алгоритма работы ЦУУ. Ввод директивы СТАРТ в этом режиме приводит к выполнению одной микрокоманды. При этом на экран терминала выводится информация о следующей микрокоманде: номер микрокоманды и содержание микрооперации. Приостановка в выполнении микропрограммы не приводит к переключению ЭВМ в состояние ОСТАНОВ. В режиме ТАКТ переход ЭВМ из состояния РАБОТА в состояние ОСТАНОВ происходит в тех же ситуациях, что и в режиме АВТ.

Для выполнения последовательности микрокоманд повторять директиву СТАРТ можно посредством ввода пустой строки, т.е. символа  $\emptyset$ . В случае необходимости наблюдения работы ЦУУ на определенном участке микропрограммы можно запустить ЭВМ в режиме АВТ или ЦИКЛ, предварительно задав останов по адресу микрокоманды, которая является первой микрокомандой исследуемого участка микропрограммы. Затем установить режим ТАКТ и выполнить необходимое количество микрокоманд. Выполнение микропрограммы в этом случае будет приостанавливаться в тех циклах работы ЦУУ, где выполняется микрокоманда, по которой задан останов.

### ***3.4. Средства контроля знаний студентов***

В целях активизации познавательной деятельности студентов в лабораторный практикум введены средства для работы с контролем и работы с моделью «неисправной» ЭВМ.

Работа с контролем устанавливается преподавателем с помощью специальных средств и может быть использована студентами в целях самоконтроля. Работа с контролем осуществляется в режиме АВТ, при этом выполнение микропрограммы приостанавливается перед исполнением каждой микрокоманды. Студент должен предсказать эту микрокоманду. Если предсказание

верно, микрокоманда выполняется, в противном случае студент может повторить попытку предсказания, но специальный счетчик ошибок, индицируемый на экране терминала, зафиксирует ошибку.

В процессе работы с контролем программа взаимодействия с пользователем, запрашивая предсказание следующей микрокоманды, осуществляет вывод на экран терминала текста «РЕГИСТР ИЛИ МК:». В ответ студент должен ввести предсказываемую микрокоманду.

Предсказание микрокоманд, символьская запись которых содержит знак присваивания, выполняется в 2 этапа: ввод левой части, ввод  $\dot{\zeta}$ ; ввод правой части, ввод  $\dot{\zeta}$ . В зависимости от вида контроля, установленного преподавателем, предсказание микрокоманд может осуществляться в различной форме.

**Контроль 1.** Ввод левой части может осуществляться как в символьном виде, так и номером микрокоманды, указанным на граф-схеме микропрограммы. Правая часть может вводиться в символьном виде или указанием числового значения выражения, указанного в правой части микрокоманды.

**Пример 3.3.** Предположим, что следующей по порядку должна выполняться микрокоманда МК7: СЧАК:=СЧАК+1, и содержимое СЧАК равно 13. В этом случае допустимым является следующий диалог:

SIM> РЕГИСТР ИЛИ МК: СЧАК

SIM> СЧАК:= I4

МИКРОМАНДА ВЫПОЛНЕНА

SIM> РЕГИСТР ИЛИ МК:

**Контроль 2.** Ввод левой части разрешен только в символьном виде, правой – как в символьном виде, так и числовым значением.

**Контроль 3.** Ввод левой части разрешен только в символьном виде, правой – только числовым значением.

Предсказание микрокоманд, не содержащих знака присваивания, выполняется подобно предсказанию левой части микрокоманд общего вида.

*Примечание.* Для того, чтобы узнать, в какой форме следует выполнять предсказание микрокоманд, необходимо после вывода моделью текста «SIM> РЕГИСТР ИЛИ МК:» ввести «? ?».

Работа с моделью «неисправной» ЭВМ позволяет проверить знание микропрограммы ЦУУ и привить студентам простейшие навыки определение некоторых видов неисправности ЭВМ. Преподавателем могут быть введены в модель ЭВМ неисправности в работе УА, АУ, неисправности регистров ЭВМ и линий передачи информации между регистрами.

Неисправности в работе УА могут проявляться в том, что не выполняется некоторая микрокоманда в микропрограмме ЦУУ или выполняются переходы, не предусмотренные микропрограммой.

Неисправности в АУ являются причиной ошибок в некотором разряде или во всех разрядах результата АОП.

Неисправности регистров ЭВМ приводят к появлению ошибок при передаче информации из регистра или в регистр и при индикации содержимого регистра.

Неисправности линий передачи приводят к блокировке передачи информации из одного регистра в другой.

*Примечание.* Рекомендуемая последовательность поиска неисправности: в режиме ЦИКЛ находится команда, при исполнении которой появилась неисправность; дальнейший поиск внутри машинного цикла данной команды ведется в режиме ТАКТ.

### **3.5. Лабораторная работа №4. Изучение основного машинного цикла**

Процесс выполнения команды или основной машинный цикл определяется микропрограммой ЦУУ (см. рис. 3.1). Предлагается изучить его на примере выполнения команд ввода-вывода. Команды ввода-вывода принадлежат к типу 1 (безадресных команд), поэтому при их наполнении не осуществляется

формирование ИА. Это приводит к тому, что машинный цикл укорачивается до 5 тактов, в течение каждого из которых выполняется одна МК. Рассмотрим детально процессы выборки команды и выполнения операций ввода-вывода (ОВВ).

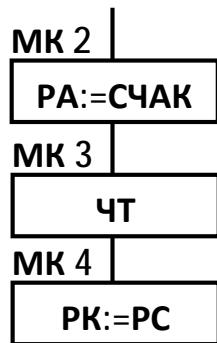


Рис. 3.2.

Алгоритм выборки команды: показанный на рис. 3.2, представляет собой последовательность трех микроопераций. Эти микрооперации: РА:=СЧАК, ЧТ, РК:=PC относятся к группе микроопераций передачи. При выполнении микрооперации ЧТ осуществляется передача содержимого ячейки ОП с адресом из РА в РС. Иначе эта микрооперация будет записана следующим образом: РС:= ОП [РА]. По окончании выборки команды, код команды, адрес которой содержится в СЧАК, записан в РК.

После выборки команды микрооперацией СЧАК:=СЧАК+1 подготавливается адрес следующей команды и осуществляется переход к выполнению операции.

Алгоритм выполнения ОВВ показан на рис. 3.3. Микрооперации АК:= РВВ и РВЫВ:=АК являются микрооперациями передачи. Логическое условие ВВ = 1, если КОП = 01, при других значениях КОП ВВ = 0. Так как проверка условия ВВ осуществляется только в случаях, если КОП = ОВВ, ВВ = 0 означает, что выполняется команда вывода.

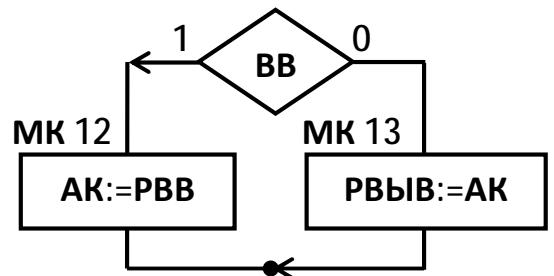


Рис. 3.3.

## ЗАДАНИЕ

1. Записать в ОП ЭВМ программу:

Адрес	Символическая команда	Код	Комментарий
	КОП	ТА	команды
00	ВВ	-	01000 AK:= PVB
01	ВЫВ	-	02000 РВЫВ:=АК
02	ОСТ	-	12000 ОСТАНОВ

2. Записать вводимое число в РВВ.

3. Убедиться в том, что программа записана в ОП без ошибок, и проверить ее в режиме ЦИКЛ.

4. Пользуясь микропрограммой ЦУУ (см. рис.3.1 – 3.3), составить для данной программы таблицу состояний ЭВМ (см. приложение 3), записав в нее ожидаемое содержимое регистров перед выполнением каждой МК.

5. Работая с контролем, проверить составленную в п.4 таблицу состояний.

6. Найти неисправность, введенную преподавателем в модель ЭВМ.

### Содержание отчета

1. Граф-схема микропрограммы ЦУУ с детализацией выборки команды и выполнения ОВВ.

2. Программа.

3. Результаты выполнения программы в режиме ТАКТ (приложение 3).

4. Описание найденной неисправности.

## 3.6. Лабораторная работа №5. Изучение микропрограммы арифметических операций

Арифметические команды, относящиеся к командам 2-ого типа (адресным командам), выполняются по полному машинному циклу, состоящему из четырех этапов (см. рис. 3.1). Выборка команды и подготовка адреса следующей команды исполняются так же, как и для команд ввода-вывода (см. 3.5). Рассмотрим детально этапы формирования ИА и выполнения арифметических операций (АОП).

Формирование ИА выполняется только для команд с косвенной или относительной адресацией, так как в случае прямой адресации ИА равен содержимому адресной части команды

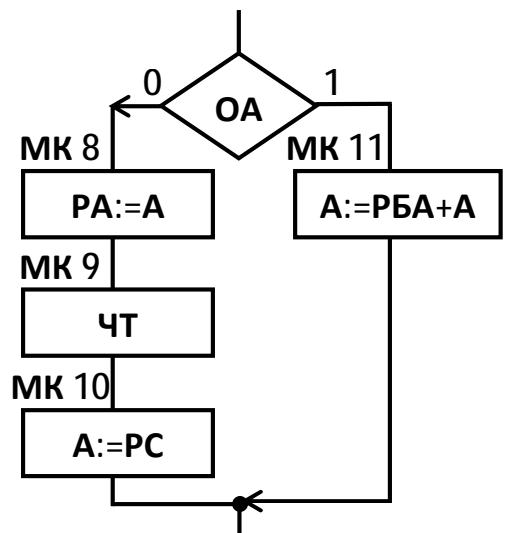


Рис. 3.4.

и после выборки команды уже записан в поле адреса РК, а в случае непосредственной адресации ИА не используется и в поле адреса РК содержится операнд.

Алгоритм формирования ИА показан на рис. 3.4. Микрооперации РА:= А, ЧТ, А:= РС относятся к группе микроопераций передачи и в случае косвенной адресации обеспечивают извлечение из ОП исполнительного адреса и запись его в поле адреса РК.

Микрооперация А:= РБА+А является микрооперацией сложения и в случае относительной адресации обеспечивает вычисление ИА.

Логическое условие ОА имеет значение 1, если ТА = 3, при других значениях ТА условие ОА равно 0. Так как проверка условия ОА выполняется только в случае, если ТА = ОА  $\vee$  КА, равенство ОА нулю означает, что адресация косвенная.

Рассмотрим алгоритм выполнения АОП, показанный на рис. 3.5.

Микрооперация РА:=А, ЧТ, РО:=РС относятся к группе микроопераций передачи и обеспечивает подготовку операнда в РО в случаях прямой, косвенной и относительной адресации. Микрооперация РО:=А также является микрооперацией передачи и обеспечивает подготовку операнда в РО в случае непосредственной адресации. Микрооперация ПУСК АУ относится к группе микроопераций установки и осуществляет установку АУ в состояние выполнения АОП.

Выполнение АОП над содержимым АК и РО и запись результата в АК происходит за один такт работы ЦУУ. В конце этого такта АУ переходит в состояние ожидание следующего сигнала ПУСК АУ. Логическое условие НА имеет значение 1, если ТА = 1, при других значениях ТА условие НА равно нулю.

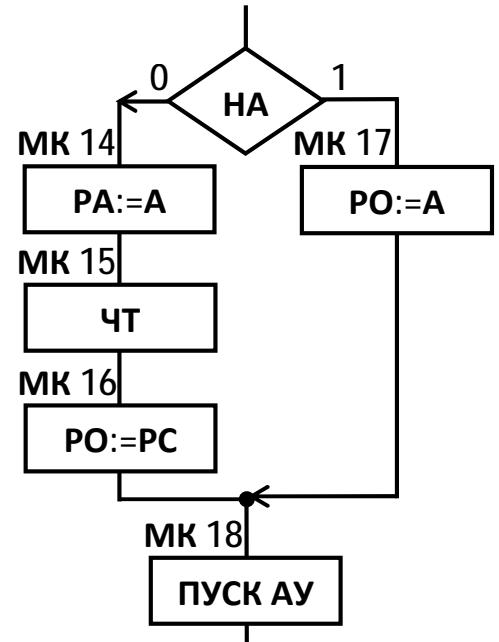


Рис. 3.5.

Особо отметим, что результат арифметической операции, находящийся в АК, сопровождается набором логических условий – так называемых признаков результата. Признак НУЛЬ указывает на равенство нулю модуля результата, признак ЗНАК указывает на отрицательный и отличный от нуля результат, признак ПП свидетельствует о возникшем в ходе операций СЛ, ВЫЧ или УМН переполнении разрядной сетки либо о попытке деления на нуль в операции ДЕЛ.

## ЗАДАНИЕ

1. Записать в ОП ЭВМ программу:

Символическая команда			Код команды	Комментарий
Адрес	КОП	ТА	А	
00	СЛ	Н	99	АК:=АК+99
01	ДЕЛ	П	05	АК:=АК/3
02	ВЫЧ	К	06	АК:=АК- ОП[ОП+[06]]
03	УМН	О	01	АК:=АК*ОП[РБА+01]
04	ОСТ		12000	ОСТАНОВ
05			3	Константа
06			07	Адрес
07			30	Константа

2. Записать базовый адрес 05 в РБА.

3. Убедиться в том, что программа записана в ОП без ошибок, осуществить СБРОС ЭВМ и проверить программу в режиме ЦИКЛ.

4. Пользуясь микропрограммой ЦУУ (рис. 3.1 – 3.5), составить для данной программы таблицу состояний ЭВМ (см. приложение 3), записав в нее ожидаемое содержимое регистров перед выполнением каждой МК.

5. Работая с контролем, проверить составленную в п.4 таблицу состояний.

6. Найти неисправность, введенную преподавателем в модель ЭВМ.

## Содержание отчета

1. Граф-схема микропрограммы ЦУУ с детализацией выборки команды, формирования ИА и выполнения АОП.

2. Программа.
3. Таблица состояний ЭВМ (см. приложение 3).
4. Описание найденной неисправности.

### **3.7. Лабораторная работа №6. Изучение микропрограммы посыльочных операций**

Выполнение посыльочных команд осуществляется в соответствии с микропрограммой ЦУУ (см. рис. 3.1). выборка команды, подготовка адреса следующей команды и формирование ИА происходит также как и при выполнении арифметических команд.

Рассмотрим алгоритм выполнения ПО, показанный на рис. 3.6. Последовательность микроопераций  $PA := A$ ,  $PC := AK$ , ЗП, которые относятся к группе микроопераций передачи, обеспечивают запись содержимого АК в ячейку ОП с ИА. Микрооперация ЗП по аналогии с ЧТ может быть записана как ОП  $[PA] := PC$ .

Последовательность микроопераций  $PA := A$ , ЧТ,  $AK := PC$  обеспечивает выполнение операции чтения в случаях прямой, косвенной и относительной адресации.

Микрооперации  $AK := 0$ ,  $PO := A$ ,  $KOP := 03$ , ПУСК АУ осуществляет выполнение операции чтения в случае непосредственной адресации. Микрооперации  $AK := 0$ ,  $KOP := 03$ , ПУСК АУ являются микрооперациями установки;  $PO := A$  – микрооперацией передачи.

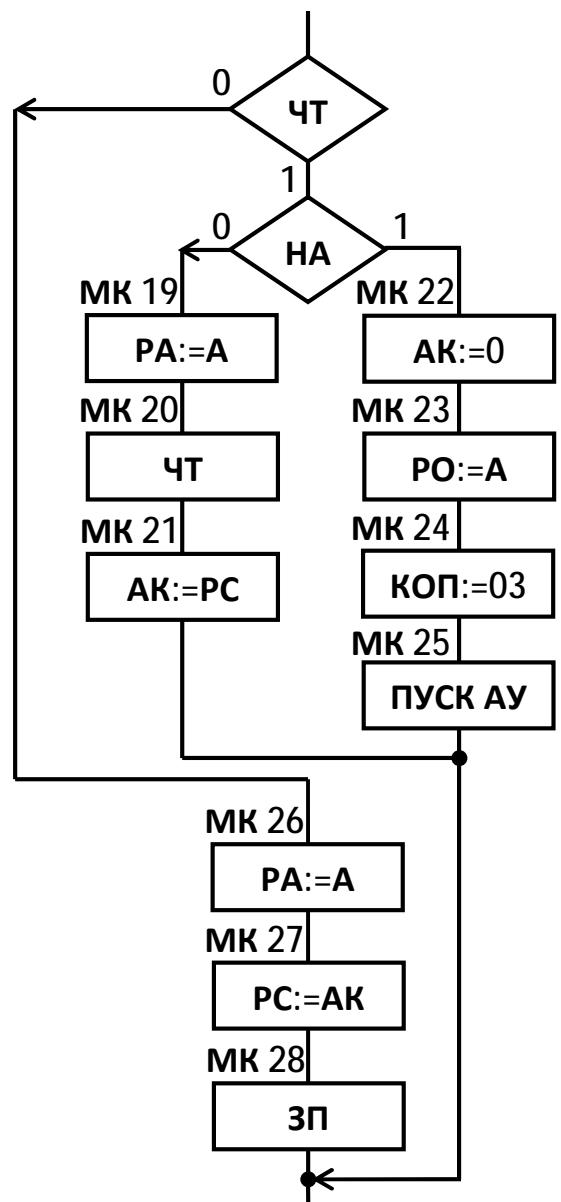


Рис. 3.6.

При непосредственной адресации чтение выполняется путем сложения операнда в РО с нулевым значением в АК. Это обусловлено невозможностью прямой передачи содержимого поля А РК в АК.

Логическое условие ЧТ имеет значение 1, если КОП=07, при других значениях КОП ЧТ=0. Так как проверка условия ЧТ осуществляется только в случаях, если КОП=ПО, ЧТ = 0 означает, что выполняется команда записи. Логическое условия НА имеет значение 1, если ТА=1, при других значениях ТА условие НА равно нулю. Следует отметить что при выполнении операции записи условие НА не проверяется, поэтому в случаях ТА=ПА и ТА=НА команда записи выполняется одинаковым образом (см. рис. 3.I).

## ЗАДАНИЕ

1. Записать в ОП ЭВМ программу:

<b>Адрес</b>	<b>Символическая команда</b>			<b>Код команды</b>	<b>Комментарий</b>
	<b>КОП</b>	<b>ТА</b>	<b>А</b>		
00	ЧТ	Н	99	07199	АК:=99
01	ЗП	П	07	08007	ОП [07] := АК
02	ЧТ	К	06	07206	АК:=ОП[ОП+[06]]
03	ЗП	О	01	08301	ОП [РБА+01]:=АК
04	ЧТ	П	07	07007	АК:= ОП[07]
05	ОСТ	-	-	12000	ОСТАНОВ
06				07	Адрес
07					Рабочая ячейка

2. Записать базовый адрес 06 в РБА.

3. Убедиться в том, что программа записала в ОП без ошибок и проверить ее в режиме ЦИКЛ.

4. Пользуясь микропрограммой ЦУУ (рис. 3.1 – 3.3, 3.6), составить для данной программы таблицу состояний ЭВМ (см. приложение 3), записав в нее ожидаемое содержимое регистров перед выполнением каждой МК.

5. Работая с контролем, проверить составленную в п.4 таблицу состояний.

6. Найти неисправность, введенную преподавателем в модель ЭВМ.

### Содержание отчета

1. Граф-схема микропрограммы ЦУУ с детализацией выборки команды, формирования ИА и выполнения ПО.

2. Программа.

3. Таблица состояний ЭВМ (см. приложение 3).

4. Описание найденной неисправности.

## 3.8. Лабораторная работа №7. Изучение микропрограммы операций передачи управления

Выполнение команд передачи управления осуществляется в соответствии с микропрограммой ЦУУ (см. рис. 3.1).

Выборка команды, подготовка адреса следующей команды и формирование ИА проходит также как и при выполнении арифметических и посыльных команд.

Алгоритм выполнения ОПУ показан на рис. 3.7. Микрооперация СКАЧ:=А является микрооперацией передачи и осуществляет формирование адреса следующей команды, равного ИА команды передачи управления. Логические условия УП0 и УПЗН имеют

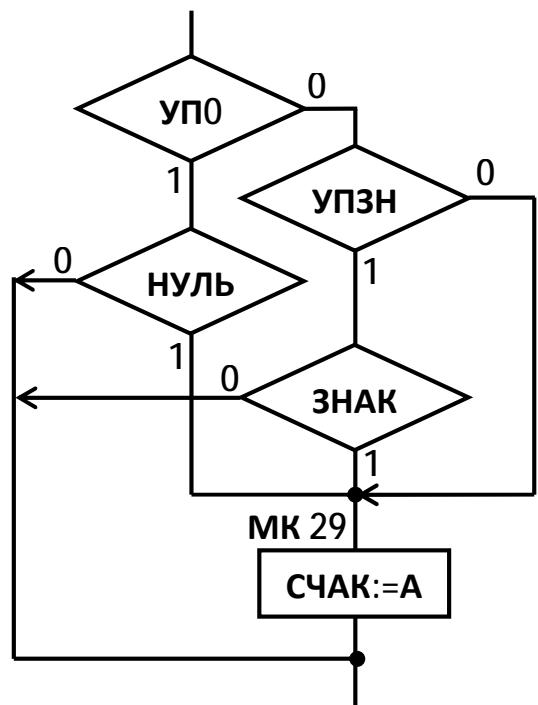


Рис. 3.7.

значения 1, если соответственно КОП = 09 и КОП = 10. Так как проверка этих условий осуществляется только в случаях, если КОП = ОПУ, УП0= 0 и одновременно УПЗН=0 означает, что выполняется команда безусловного перехода. Логические условия НУЛЬ и ЗНАК равны 1 соответственно при нулевом и отрицательных значениях аккумулятора. Таким образом, изменение содержимого СКАЧ при выполнении команды условного перехода по нулю происходит, если НУЛЬ=1; при выполнении условного перехода по знаку – если ЗНАК=1; при выполнении безусловного перехода – независимо от значений НУЛЬ или ЗНАК.

Следует отметить, что команда передачи управления в случаях ТА=ПА и ТА=НА выполняются одинаковым образом (рис.3.1).

### ЗАДАНИЕ

1. Записать в ОП ЭВМ программу:

<b>Адрес</b>	<b>Символическая команда</b>			<b>Код команды</b>	<b>Комментарий</b>
	<b>КОП</b>	<b>ТА</b>	<b>А</b>		
00	ЧТ	Н	00	07000	АК:=0
01	УПЗН	П	04	10004	СКАЧ:= 04, если АК<0
02	УП0	К	09	09209	СКАЧ:= 05, если АК=0
03	УПЗН	П	07	10007	СКАЧ:= 07, если АК<0
04	ОСТ			12000	ОСТАНОВ
05	ВЫЧ	Н	06	04106	АК:=АК-6
06	БП	0	02	11302	СЧАК:=РБА+02
07	СЛ	Н	08	03108	АК:=АК+8
08	БП	П	02	11002	СЧАК:=02
09				05	Адрес

2. Записать базовый адрес 00 в РБА

3. Убедиться в том, что программа записана в ОП без ошибок и проверить ее в режиме ЦИКЛ.

4. Пользуясь микропрограммой ЦУУ (рис. 3.1–3.3, 3.5–3.7), составить для данной программы таблицу состояний ЭВМ (см. приложение 3), записав в нее ожидаемое содержимое регистров перед выполнением каждой МК.

5. Работая с контролем, проверить составленную в п.4 таблицу состояний.

6. Найти неисправность, введенную преподавателем в модель ЭВМ.

### Содержание отчета

1. Граф-схема микропрограммы ЦУУ с подробным изображением выборки команды, формирование ИА и выполнения ОПУ.

2. Программа.

3. Таблица состояний ЭВМ (см. приложение 3).

4. Описание найденной неисправности.

## **4. Управление виртуальной ЭВМ**

Ввод каждой директивы или значения параметра заканчивается нажатием  $\zeta$ ; пока клавиша  $\zeta$  не нажата, ошибку ввода можно исправить, нажимая <Backspace>.

### **4.1. Директивы управления системой**

<?> – запрос подсказки (список директив или действующее значение параметра);

< $\zeta$ > (пустой ввод) – зарезервирован для дальнейшего использования;

<П> – ввести параметры модели (подробнее см. п.4.2);

<СБРОС> – сбросить счетчики, "Контроль", "Неисправность", снять блокировку модели по счетчику ошибок;

<K> – установить/снять контроль. После появления "=" введите предельное число попыток для ответа. Ввод K $\leq$ 0 снимает контроль, при K $\geq$ 7 количество попыток не ограничивается (продолжение работы только при правильном ответе). При установке контроля автоматически снимается неисправность.

<Н> – ввести код неисправности. После появления "=" введите код (от 1 до 5 символов). При установке неисправности контроль автоматически снимается. Логический смысл неисправности не проверяется.

### **4.2. Параметры модели**

Переустанавливаются по директиве <П>. В ответ на вопрос системы вводится новое значение. При вводе <?> выводится справка о действующем значении. Ввод < $\zeta$ > (пустой ввод) сохраняет действующее значение параметра. Если введено значение вне диапазона, будет установлено граничное значение.

**<КВАНТ: (Предельное к-во тактов до останова (10-9999):>** – запрашивается количество микроопераций, после выполнения которого происходит останов модели **По истечению времени**;

**<Инициализировать схему (0-нет, 1-да):>** – при установке 1 работа модели в режимах "Авт." и "Цикл" сопровождается текущей индикацией изменяемого содержимого регистров схемы. При установке 0 индикация выполняется один раз при выходе на останов (модель работает быстрее);

**<Предел для счетчика ошибок в режиме "КОНТРОЛЬ (0-128):>** – запрашивается количество ошибок, по достижении которого работа модели блокируется. При установке значения 0 блокировки не будет при любом количестве ошибок;

**<Уровень контроля: (1-3):>** – уровень 1 (мягкий) разрешает ввод вместо левой части микрокоманды (МК) ее номер, уровень 2 (средний) запрещает ввод номера МК в левой части, оставляя возможность ввести правую часть МК текстом, уровень 3 (жесткий) разрешает ввод левой части МК только текстом, правой – числом;

**<Режим индикации МК (1 – да, 2 – только текст, 3 – нет):>** – в режиме 1 на индикацию выводится номер МК и ее текст, в режиме 2 – только текст, в режиме 3 – индикация отсутствует;

**<Рабочий каталог:>** – по умолчанию рабочий каталог для хранения файлов программ имеет имя LAB1. В ответ на запрос можно изменить имя рабочего каталога;

**<Авторежим:>** – значение 11 устанавливает режим автоматического сохранения/восстановления программы при выходе из модели и при входе в нее; значение 10 устанавливает режим автоматического сохранения программы, значение 1 режим автоматического восстановления, 0 – отменяет автоматические режимы.

#### ***4.3. Кодирование неисправностей***

Неисправность кодируется номером (0-8) и одним или двумя параметрами.

**Пример:** K=12345 (1 – номер неисправности, 23 – параметр 1, 45 – параметр 2).

В качестве параметров указывается: номер микрокоманды (МК=0-31), номер регистра (RG=1-30) или разряда (Р=0-5, причем 5 означает знаковый разряд, 1 – младший разряд, 0 – все разряды).

Используются следующие внутренние номера регистров модели, имеющихся на схеме ЭВМ:

**пятиразрядные:** 1 – АК, 2 – РО, 3 – РС, 4 – память (вся), 5 – РК (весь), 9 – РВВ, 10 – РВЫВ;

**двуразрядные:** 11– РА, 12 – А (часть РК), 13 – РБА, 14 – СЧАК;

**одноразрядные:** 21 – КОП (часть РК), 22 – ТА (часть РК), 24 – НУЛЬ, 25 – ЗНАК, 30 – ПП.

*Коды неисправности:*

**0** – нет неисправности (параметры не используются);

**1 МК** – не выполняется микрокоманда с номером "МК";

**2 МКА МКВ** – после микрокоманды "МКА" всегда следует микрокоманда "МКВ";

**3 МК Р** – выпадение (установка в 0) разряда "Р" из результата микрокоманды "МК";

**4 RG Р** – то же, при индикации содержимого регистра "RG" на схеме;

**5 RG Р** – то же, при записи в регистр "RG" (микрокомандой или директивой);

**6 RG Р** – то же, при чтении из регистра "RG" (в том числе и по директиве "?");

**7 RGA RGB** – не передается информация из регистра "RGB" в регистр "RGA";

**8 ТИП** – ошибка в АУ (тип 1 – при вычитании, 2 – умножении, 3 – делении, 4 – сложении);

**9 RG** – инвертирование регистров НУЛЬ, ЗНАК и ПП.

---

<sup>1</sup> Вход в режим управления осуществляется директивой УПР, выход – ВЫХ. Password сообщается при регистрации на сайте программы.

## 5. Сети коммуникаций

### 5.1. Маршрутизация

*Маршрут* – список узлов коммутации, каналов и линий связи в сети, от узла отправителя к узлу получателю.

**Пример.** Пусть задана сеть (рис. 5.1), в которой узлы между собой связаны линиями связи. Существует необходимость передачи данных от 3 узла к 7. Узел 3 будем называть отправителем, узел 7 – получателем. Информация передается через транзитные узлы, в данном случае такими узлами выступают узлы 2-6, *маршрутом* же будет выступать последовательность узлов 3-2-6-7.

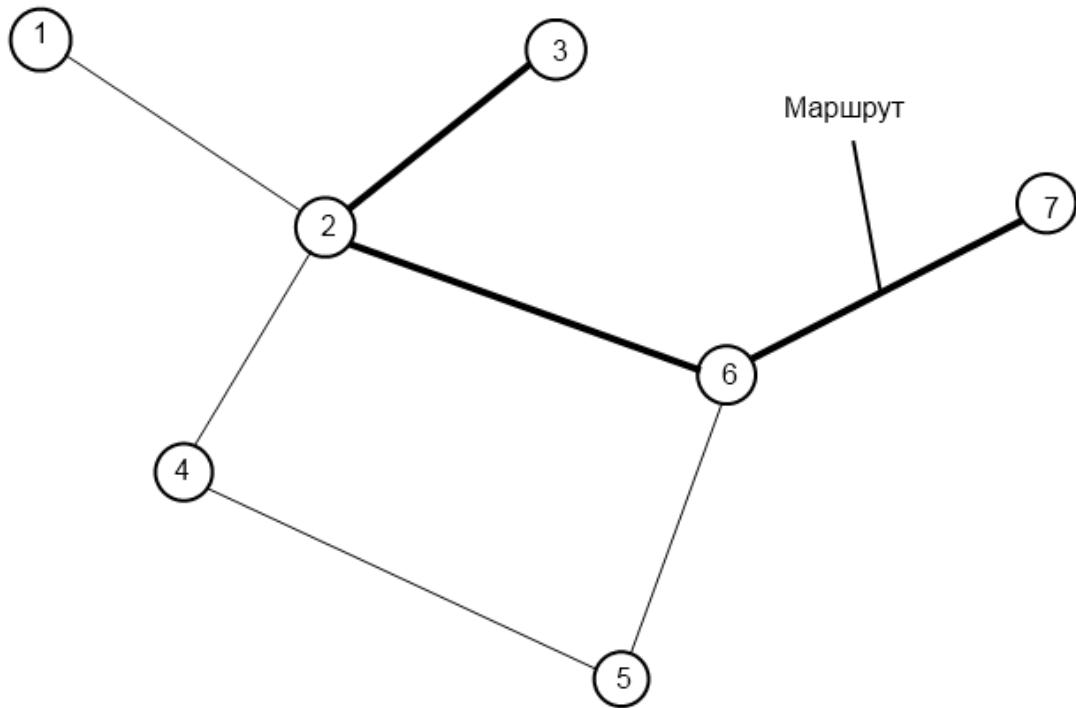


Рис 5.1. Структура сети

*Маршрутизация* – процедура, определяющая оптимальный по заданным параметрам маршрут в сети и связи между узлами коммутации.

При выборе маршрута иногда ограничиваются только информацией о топологии сети.

При топологии сети, заданной на рисунке 5.2, между узлами 1-4 существует несколько альтернативных маршрутов:

1-4

1-2-4

1-3-4

Если учесть только топологию, то целесообразно выбрать маршрут, который имеет меньше транзитных узлов – маршрут 1-4.

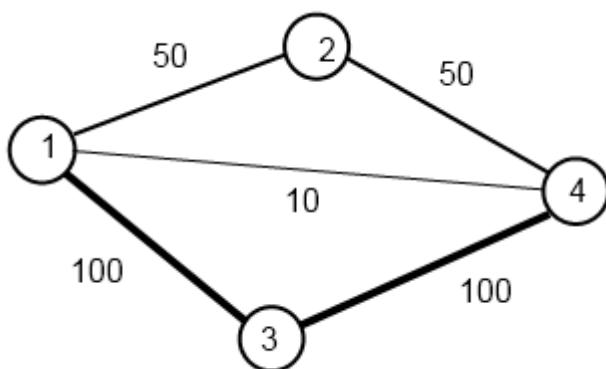


Рис 5.2. Выбор маршрута с учетом только топологии сети

Однако, если учесть пропускную способность сети, то выбор маршрута будет уже не таким очевидным.

На рисунке 5.2 показано, что каналы 1-2, 2-4 имеют пропускную способность 50 Мбит/с, каналы 1-3, 3-4 – 100 Мбит/с, а канал 1-4 – 10 Мбит/с. Если критерием при выборе маршрута будет максимальная скорость передачи данных, то оптимальным будет маршрут 1-3-4, хотя он проходит через большее число узлов.

Абстрактный способ измерения степени близости между двумя объектами называется *метрикой*.

Для построения метрики, учитывающей пропускную способность, длину каждого канала-участка характеризуют величиной, обратной его пропускной способности. Чтобы оперировать целыми числами, выбирают некоторую константу, заведомо большую, чем пропускные способности каналов в сети или

равную максимальной пропускной способности всех каналов, или же наименьшее общее кратное пропускных способностей всех каналов.

Если в качестве константы возьмем 100 Мбит/с, то метрика каналов 1-2, 2-4 будет равна 2, каналов 1-3, 3-4 – 1, а канала 1-4 – 10.

*Метрика маршрута* равна сумме метрик составляющих его каналов.

Метрика каждого из возможных маршрутов будет равна соответственно:

для маршрута 1-4=10;

для маршрута 1-2-4=4;

для маршрута 1-3-4=2.

Выбрав самый короткий маршрут с метрикой 2, получим оптимальный маршрут: 1-3-4.

Следует, однако, упомянуть тот факт, что при выборе, какого либо маршрута необходимо учитывать так же загруженность сети на данный момент, а так же критерий поиска маршрута.

## **5.2. Лабораторная работа №8. Поиск оптимального маршрута по критерию пропускной способности коммуникационной сети**

**Пример 5.1.** Пусть задана топология сети (рис. 5.3):

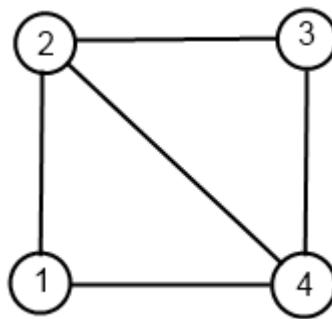


Рис 5.3. Топология сети

Для реализации маршрутизации сети в каждом транзитном узле коммутации, начиная с узла *источника*, формируется *таблица маршрутизации*, которая представляет собой матрицу, размерностью  $(U - 1) \times L_j$ :

$$M^j = \left\| m_{iv}^{(j)} \right\|_{(U-1) \times L_j} = \left( \overline{m_1^{(j)}}, \dots, \overline{m_i^{(j)}}, \dots, \overline{m_{j-1}^{(j)}} \overline{m_{j+1}^{(j)}}, \dots, \overline{m_U^{(j)}} \right), \quad (1)$$

$$\overline{m_i^{(j)}} = \left( \overline{m_1^{(j)}}, \dots, \overline{m_{iv}^{(j)}}, \dots, \overline{m_{iL_j}^{(j)}} \right); v = \overline{1, L}; i, j = \overline{1, U}; i \neq j \quad (2)$$

где  $U$  – количество узлов коммутации в сети;  $L_j$  – количество исходящих линий связи из  $j$ -го узла коммутации.

Матрица  $\vec{I}^{i,j}$  определяет, какую **исходящую линию связи** предпочтительно выбрать из  $j$ -го узла коммутации при поиске маршрута к  $i$ -му узлу (узлу получателю).

Первый элемент  $m_{i1}^{(j)}$  вектор-строки формулы (2) указывает номер исходящей линии связи из  $j$ -го узла коммутации **к смежному узлу коммутации**, которую предпочтительнее выбрать для организации маршрута к  $i$ -му узлу коммутации (узлу получателю).

Второй элемент (2) указывает номер следующей исходящей линии связи **из  $j$ -го узла коммутации к другому смежному узлу коммутации**, которая менее предпочтительна для организации искомого маршрута. И так до  $L_j$ -го элемента вектор-строки (2).

В данном случае:  $m_{il}^{(j)}$  – исходящая линия связи выбора  $L$  смежных линий связи из  $j$ -го узла.

**Пример 5.2.** Пусть необходимо построить таблицу маршрутизации для 4-го узла коммутации, если топология сети задана на рис 5.3.

Строки матрицы маршрутизации соответственно будут равны:

$$\overline{m_1^{(4)}} = (1, 2, 3); \overline{m_2^{(4)}} = (2, 3, 1); \overline{m_3^{(4)}} = (3, 1, 2);$$

При поиске маршрута от узла коммутации 4 к узлу коммутации 1 необходимо обратиться к вектор-строке  $\overline{m_1^{(4)}} = (1, 2, 3)$ . Исходящая линия связи к узлу коммутации 1 является более предпочтительной, так как она ведет непосредственно к искомому УК, следовательно, является исходящей линией связи

первого выбора. Соответственно, исходящие линии связи к узлам 3 и 2 являются исходящими линиями связи второго и третьего выбора.

Для того, чтобы была возможность определять маршруты между любой парой узлов коммутации необходимо построить таблицы маршрутизации в каждом узле сети.

Совокупность таблиц маршрутизации для всех УК называется *планом распределения информации* в сети.

**Пример 5.3.** Создадим *план распределения информации* для топологии сети, показанной на рисунке 5.4. Критерием выберем максимальную скорость передачи данных.

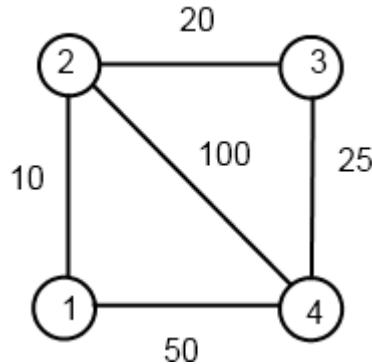


Рис 5.4. Топология сети

$$M^1 = \begin{vmatrix} 2 & 4 & 2 \\ 3 & 4 & 2 \\ 4 & 4 & 2 \end{vmatrix} \quad M^2 = \begin{vmatrix} 1 & 4 & 3 & 1 \\ 3 & 3 & 4 & 2 \\ 4 & 4 & 3 & 1 \end{vmatrix} \quad M^3 = \begin{vmatrix} 1 & 4 & 2 \\ 2 & 2 & 4 \\ 4 & 4 & 2 \end{vmatrix} \quad M^4 = \begin{vmatrix} 1 & 1 & 2 & 3 \\ 2 & 2 & 1 & 3 \\ 3 & 2 & 3 & 1 \end{vmatrix}$$

Формирование *плана распределения информации* может осуществляться и по другим критериям:

1. Количество транзитных узлов
2. Время задержки при передаче данных
3. Безопасность узлов связи
4. Надёжность узлов связи

*План распределения информации* позволяет выбрать маршрут между любой парой узлов, начиная с узла источника и заканчивая узлом получателем. Для этого

необходимо из *таблицы распределения* выбрать линии связи первого выбора, если она не доступна, то выбирают линию второго выбора и т.д. Данная процедура продолжается во всех узлах, участвующих в формировании искомого маршрута, пока не будет определен маршрут между заданной парой узлов.

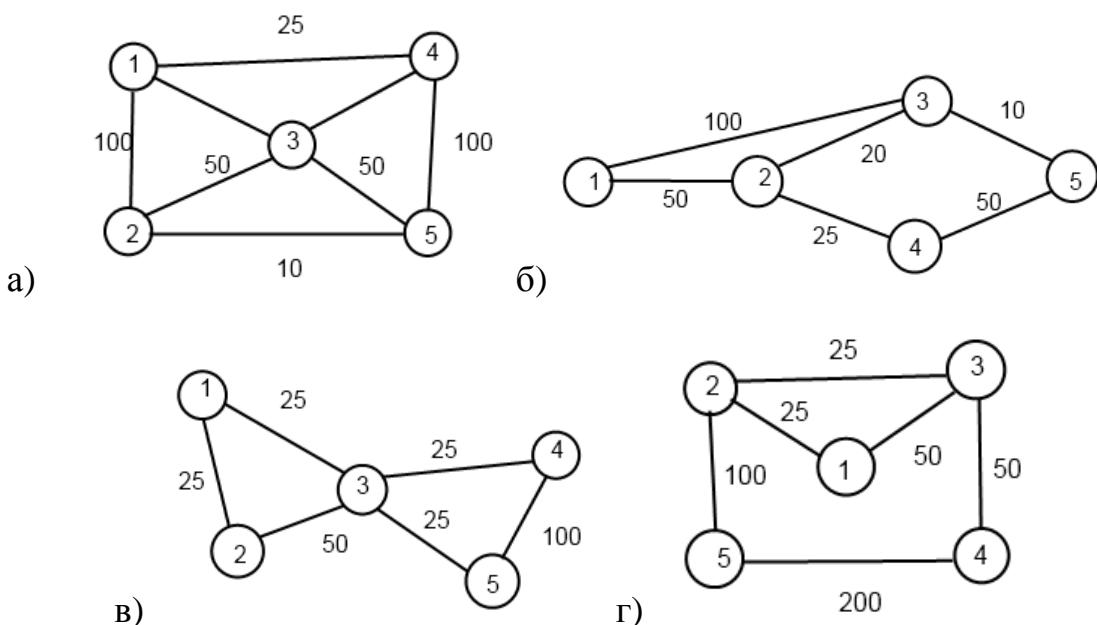
В случае недоступности всех исходящих линий связи в данном узле потребуется либо вернуться на предыдущий узел коммутации и выбрать менее предпочтительную исходящую линию связи, либо дать отказ на невозможность организации искомого маршрута между заданной парой узлов.

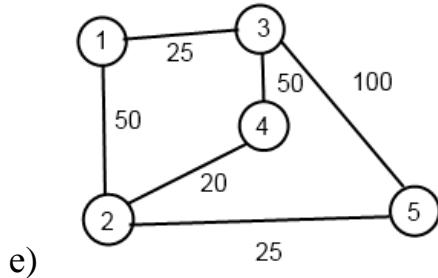
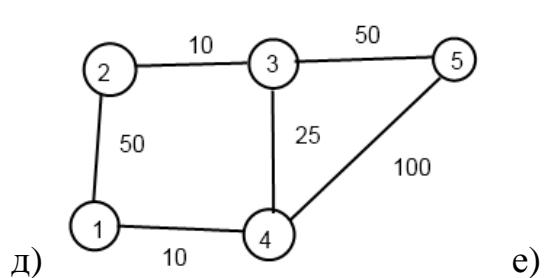
Таким образом, анализируя процедуры, участвующие в формировании маршрутов можно заключить, что **маршрутизация состоит из двух этапов:**

1. Формирование плана распределения информации в сети связи.
2. Выбор исходящих линий связи в узле коммутации при поиске маршрута между узлами отправителем и получателем.

### ЗАДАНИЕ

1. Найти оптимальный маршрут от узла 1 к узлу 5 для заданной топологии, по критерию пропускной способности сети.

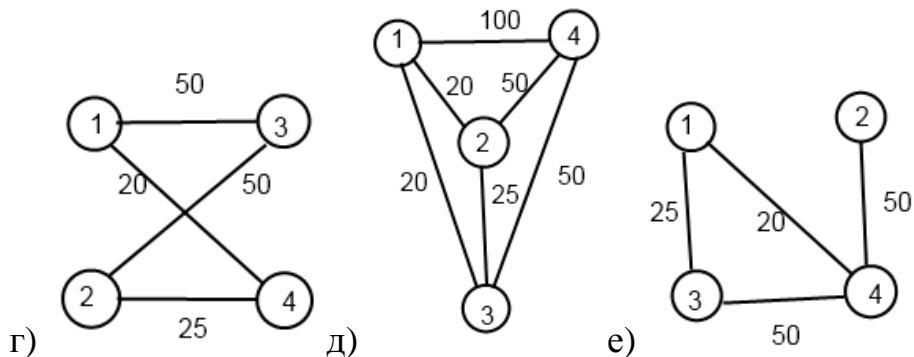
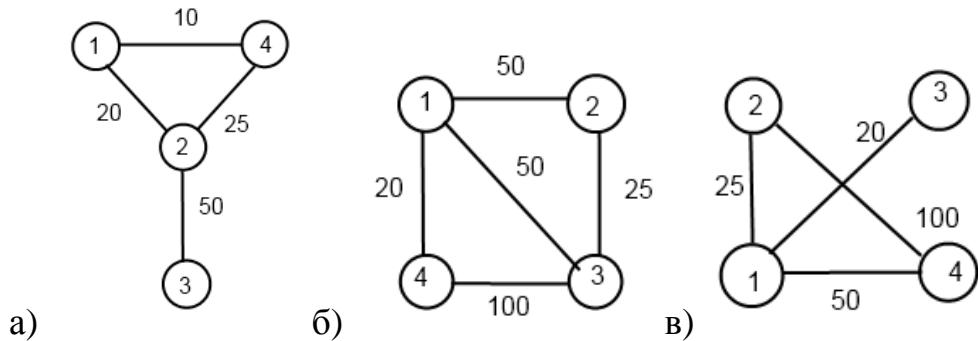




2. Сформировать план распределения информации для заданной топологии по следующим критериям:

1) количество транзитных узлов

2) скорость передачи данных.



3. Написать программу поиска оптимального маршрута в сети, реализовать поиск по критериям:

1) минимального количества узлов коммутации;

2) максимальной скорости передачи данных.

2)

## Содержание отчета

1. Оптимальный маршрут в соответствии с заданием 1.
2. План распределения информации для заданной топологии в соответствии с указанными в задании 2 критериями.
3. Листинг программы.
4. Примеры результатов работы программы для указанных в задании 3 критериев.

### **5.3. Лабораторная работа №9. Методы формирования плана распределения информации**

#### *Метод рельефов.*

Суть данного метода состоит в следующем. Пусть  $i$  – произвольный узел коммутации сети связи, тогда  $i$ -рельефом называется процедура присвоения значения числовой функции каждой линии связи. Процедуру построения  $i$ -рельефа можно описать следующим образом. Из  $i$ -го узла коммутации всем исходящим линиям связи присваивается число 1. Все узлы коммутации, в которые поступило число 1, передают по всем исходящим линиям связи, кроме тех линий связи, по которым поступило 1, число 2. Далее узлы коммутации, на которые поступило число 2, передают по линиям связи, кроме тех, по которым поступило 2, число 3 и т.д., до тех пор, пока все линии связи не будут пронумерованы. Говорят, что линия связи имеет  $n$ -ю высоту, если она обозначена числом  $n$  в  $i$ -рельефе.

Указанным способом формируется рельеф из каждого УК сети связи. В итоге получается, что каждая линия связи имеет  $S$  высот. В результате линии связи с минимальной высотой является исходящей линией связи первого выбора. Линии связи с большими высотами, соответственно, являются исходящими линиями связи второго, третьего и т.д. выбора.

**Пример 5.4.** Построим рельеф для топологии заданной на рисунке 5.5, узлом получателем в данном случае выберем узел 1. По исходящим линиям связи 1-2, 1-3, 1-4 передаётся число 1. Из узлов 2 3 4 по исходящим линиям связи 3-8, 3-4, 4-6, 4-5, 4-2, 2-5 передаётся число 2, из узлов 5 6 8 по исходящим линиям связи 6-8, 6-9, 5-7 передаётся число 3, из узла 9 по исходящим линиям 9-8, 9-7 передаётся число 4. Для нахождения кратчайшего пути из любого узла к узлу 1 необходимо выбирать линии связи с наименьшим весом. Так, например кратчайший путь из узла 6 к узлу 1 будет 6-4-1. Недостаток метода в том, что для каждого узла получателя необходимо строить рельеф.

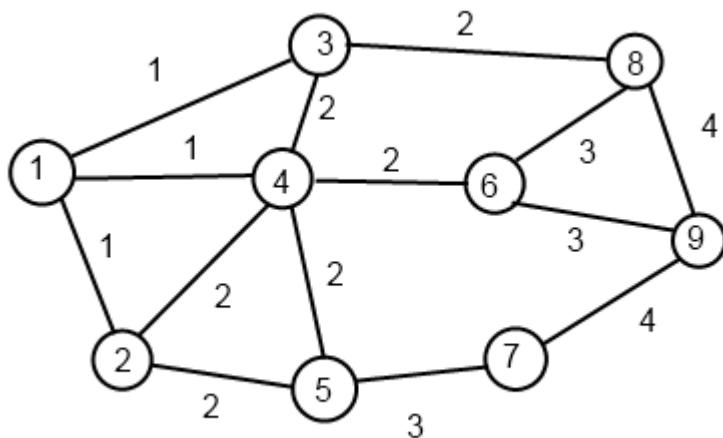


Рис 5.5. Топология сети

#### *Игровой метод.*

Изначально план распределения информации формируется в виде таблиц маршрутизации, соответственно для всех строк ставятся в соответствие весовые коэффициенты таким образом, что сумма всех весовых коэффициентов должна быть равной единице. В результате формируется матрица весовых коэффициентов. При поиске маршрута из узла источника к  $i$ -тому узлу получателю, во всех транзитных узлах коммутации происходит обращение к соответствующим строкам таблиц маршрутизации. В соответствующей строке матрицы весовых

коэффициентов, выбирается исходящая линия связи с максимальным весовым коэффициентом.

**Пример 5.5.** Составим план распределения информации игровым методом для топологии сети заданной на рисунке 5.6.

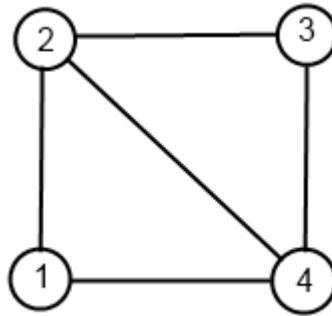


Рис 5.6. Топология сети

Пусть изначальный план распределения информации задан в виде таблиц маршрутизации. Весовые коэффициенты будут равны соответственно:

$$P^1 = (p_2^1, p_3^1, p_4^1) = \begin{array}{c|cc} & 2 & 4 \\ \hline 2 & 0,7 & 0,3 \\ 3 & 0,5 & 0,5 \\ 4 & 0,3 & 0,7 \end{array}$$

$$P^2 = (p_1^2, p_3^2, p_5^2) = \begin{array}{c|ccc} & 1 & 3 & 4 \\ \hline 1 & 0,6 & 0,1 & 0,3 \\ 3 & 0,2 & 0,6 & 0,2 \\ 4 & 0,2 & 0,2 & 0,6 \end{array}$$

$$P^3 = (p_1^3, p_2^3, p_4^3) = \begin{array}{c|cc} & 2 & 4 \\ \hline 1 & 0,5 & 0,5 \\ 2 & 0,7 & 0,3 \\ 4 & 0,3 & 0,7 \end{array}$$

$$P^4 = (p_1^4, p_2^4, p_3^4) = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0,7 & 0,2 & 0,1 \\ 2 & 0,2 & 0,6 & 0,2 \\ 3 & 0,1 & 0,2 & 0,7 \end{array}$$

Выберем исходящую линию связи из узла источника 2 к узлу получателю 1 при условии, что число транзитных узлов не должно превышать одного. Для этого из матрицы весовых коэффициентов  $P^2$  выберем строку

$$p_1 = (0,6 \ 0,1 \ 0,3).$$

Исходящая линия связи из узла 1 будет являться линией связи первого выбора, так как её весовой коэффициент является наибольшим, если данная линия связи не доступна, то будет выбрана исходящая линия связи второго выбора в данном случае линия связи из узла 4 так как её весовой коэффициент больше чем у узла 3. Предположим, что исходящая линия связи первого выбора была занята, и для организации маршрута была выбрана исходящая линия связи второго выбора. Для дальнейшей организации маршрута из матрицы весовых коэффициентов  $P^4$  выберем строку

$$p_1^4 = (0,7 \ 0,2 \ 0,1).$$

Допустим, что исходящая линия связи первого выбора доступна. Следовательно, маршрут из узла 2 к узлу 1 организован и имеет вид {2 4 1}. Линии связи, которые участвуют в маршруте, поощряются, следовательно, их весовые коэффициенты должны быть увеличены, а строки матрицы весовых коэффициентов нормированы. При увеличении весовых коэффициентов необходимо заранее выбрать константу, на которую будем увеличивать весовой коэффициент поощрённых линий. Для данного примера пусть эта константа будет равна 0,2 тогда соответствующие строки матриц весовых коэффициентов примут вид :

$$p_1^2 = (0,6 \ 0,1 \ 0,5); p_1^4 = (0,9 \ 0,2 \ 0,1).$$

Следовательно данные векторы строки необходимо нормировать. Для нормирования необходимо каждый из коэффициентов разделить на сумму модулей всех коэффициентов вектора.

После нормирования данные строки примут вид:

$$p_1^2 = (0,5 \ 0,08 \ 0,42); p_1^4 = (0,75 \ 0,17 \ 0,08).$$

При повторном поиске маршрута соответствующие коэффициенты строк матриц весовых коэффициентов изменятся и примут вид

$$p_1^2 = (0,42 \ 0,07 \ 0,51); p_1^4 = (0,79 \ 0,14 \ 0,07).$$

Видно, что исходящая линия связи из узла 2 к узлу 4 стала линией связи первого выбора при поиске маршрута к узлу 1, т.к. её коэффициент стал наибольшим.

Следовательно, матрицы весовых коэффициентов примут вид

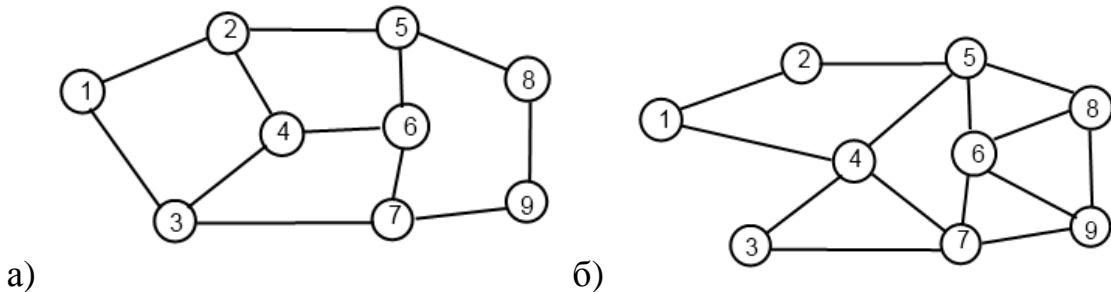
$$P^2 = (p_1^2, p_3^2, p_5^2) = \begin{matrix} & \begin{array}{ccc} 1 & 3 & 4 \end{array} \\ \begin{matrix} 1 \\ 3 \\ 4 \end{matrix} & \left| \begin{array}{ccc} 0,42 & 0,07 & 0,51 \\ 0,2 & 0,6 & 0,2 \\ 0,2 & 0,2 & 0,6 \end{array} \right| \end{matrix}$$

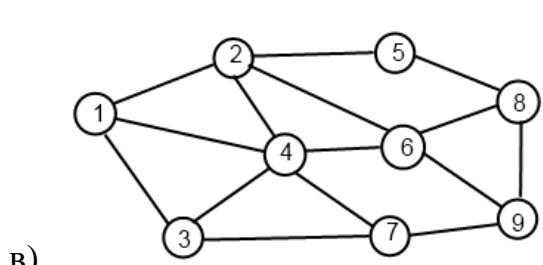
$$P^4 = (p_1^4, p_2^4, p_3^4) = \begin{matrix} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \left| \begin{array}{ccc} 0,79 & 0,14 & 0,07 \\ 0,2 & 0,6 & 0,2 \\ 0,1 & 0,2 & 0,7 \end{array} \right| \end{matrix}$$

Метод решает задачу оптимизации сети по критерию вероятности установления связи между узлами. Отсутствие необходимости передачи служебной информации является также достоинством данного метода. Однако при выходе из строя, какой либо из линий связи, необходимо переформировать план распределения информации.

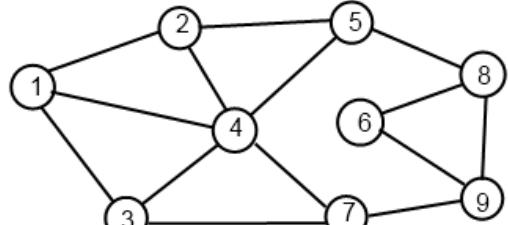
### ЗАДАНИЕ

1. Построить рельеф в заданной топологии сети для узла получателя 1.

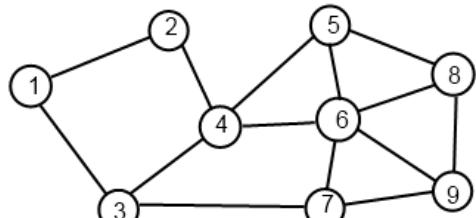




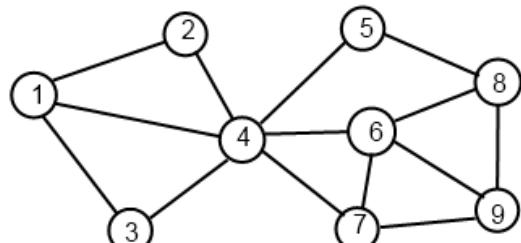
в)



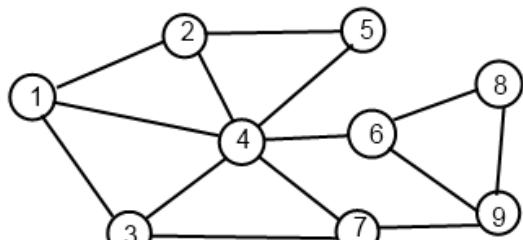
г)



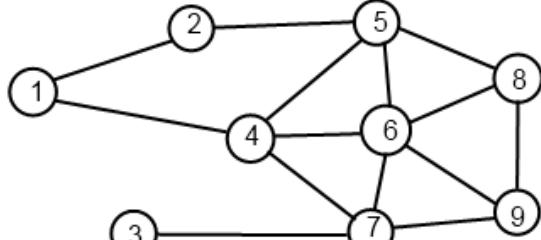
д)



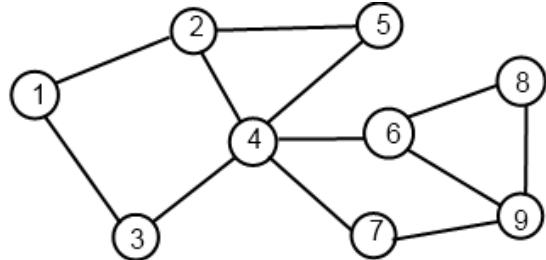
е)



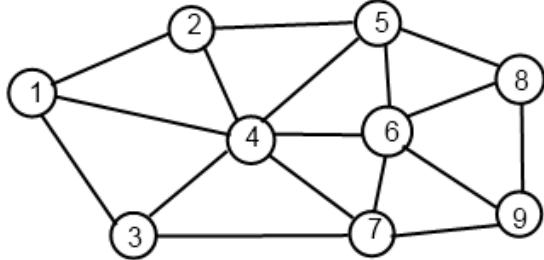
ж)



з)



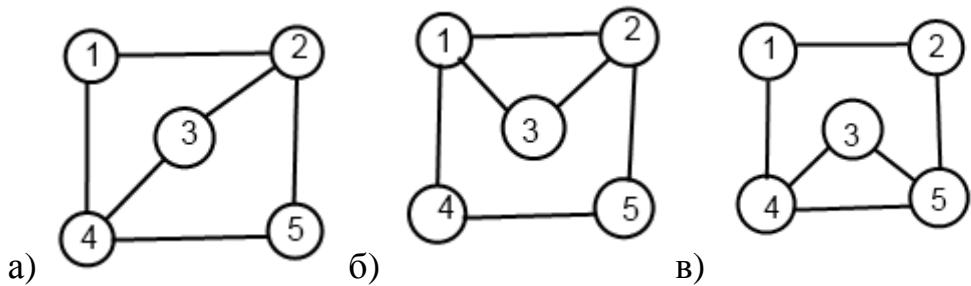
и)



к)

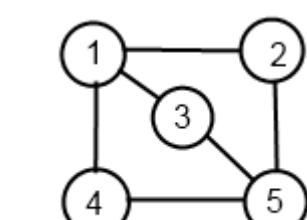
2. Написать программу для построения рельефа в заданной топологии.

3. Составить план распределения информации игровым методом, организовать маршрут между узлом источником 1 и узлом получателем 3 с количеством транзитных узлов не более 2, при условии что линия связи между узлами 4 и 5 не доступна.

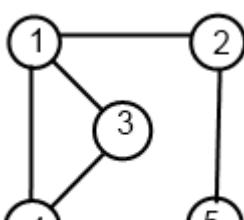


б)

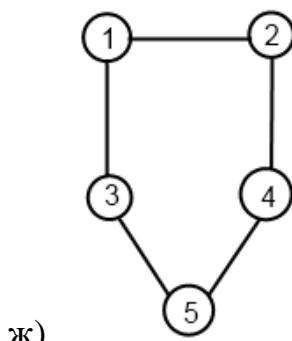
в)



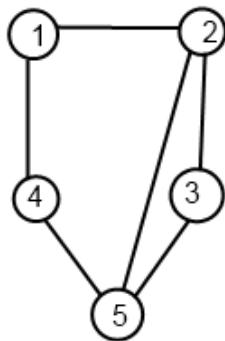
д)



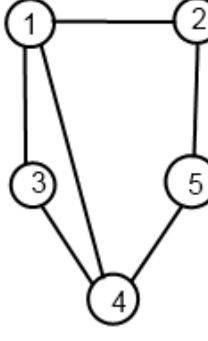
е)



з)



и)



к)

4. Написать программу для реализации игрового метода в заданной топологии.

#### Содержание отчета

1. Рельеф в заданной топологии сети для узла получателя 1 в соответствии с заданием 1.
2. Листинг программы для построения рельефа в заданной топологии.
3. Примеры результатов работы программы построения рельефа.
4. План распределения информации, составленный игровым методом, для заданной топологии в соответствии с указанными в задании 3 условиями.
5. Листинг программы реализации игрового метода в заданной топологии.
6. Примеры результатов работы программы для указанных в задании 3 критериев.

## **5.4. Лабораторная работа №10. Статистические оценки характеристик сети**

Для того, чтобы представить характеристики качества передачи последовательности пакетов через сеть в компактной форме, применяются статистические методы. Статистические характеристики выявляют закономерности в поведении сети, которые устойчиво проявляются только на длительных периодах времени.

Основным инструментом статистики является так называемая гистограмма распределения оцениваемой случайной величины. Рассмотрим этот инструмент на примере такой характеристики сети, как задержка пакета.

Будем считать, что нам удалось измерить задержку доставки каждого из 2600 пакетов, переданных между двумя узлами сети, и сохранить полученные результаты. Эти результаты называются выборкой случайной величины.

Для того чтобы получить гистограмму распределения, мы должны разбить весь диапазон измеренных значений задержек на несколько интервалов и подсчитать, сколько пакетов из нашей выборки попало в каждый интервал. Пусть все значения задержек укладываются в диапазон 20-90 мс. Разобьем его на семь интервалов по 10 мс. В каждый из этих интервалов, начиная с интервала 20-30 мс и т.д., попало 100 (n1), 200 (n2), 300 (n3), 300 (n4), 400 (n5), 800 (n6) и 500 (n7) пакетов соответственно. Отобразив эти числа в виде горизонтальных уровней для каждого интервала, мы получим гистограмму, показанную на рис. 5.7, которая, основываясь всего на семи числах n1, n2,..., n7, дает нам компактную статистическую характеристику задержек 2600 пакетов.

Гистограмма задержек дает хорошее представление о производительности сети. По ней можно судить, какие уровни задержек более вероятны, а какие — менее. Чем больше период времени, в течение которого собираются данные для построения гистограммы, тем с более высокой степенью достоверности можно предсказать поведение сети в будущем.

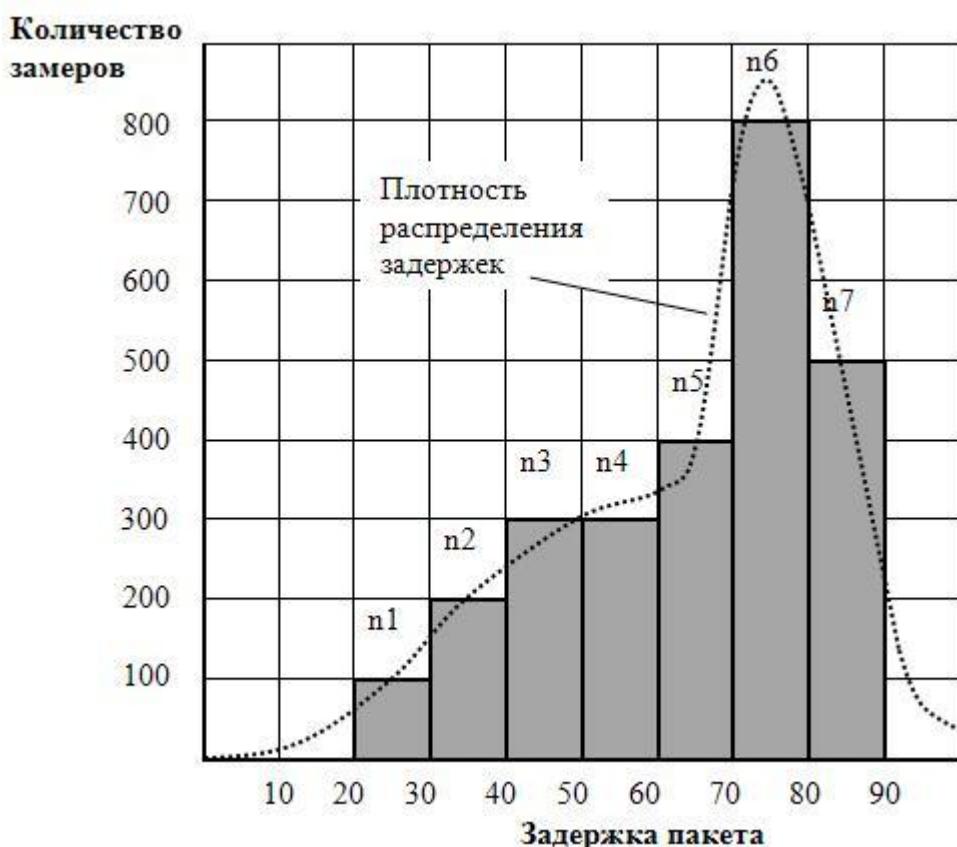


Рис. 5.7. Гистограмма распределения задержек

Например, пользуясь гистограммой на рис. 5.7, можно сказать, что и в будущем при измерениях задержек пакетов у 65 % пакетов задержка не превысит 60 мс. Для получения такой оценки мы сложили общее количество пакетов, задержки которых попали во все интервалы, большие 60 мс (1700 замеров), и разделили эту величину на общее количество пакетов (2600 замеров). Другими словами, мы нашли долю пакетов, задержки которых в выборке превышают 60 мс, и считаем, что наша выборка позволяет судить о поведении сети в будущем.

Гистограмма дает хорошее детальное описание соответствующей характеристики, но чаще всего используются еще более компактные статистические оценки характеристик, которые позволяют представить характеристику одним числом на основе некоторой математической обработки имеющейся выборки.

При увеличении количества интервалов и времени наблюдения мы в пределе получаем непрерывную функцию, которая называется плотностью распределения

задержки доставки пакета (показана пунктиром). В соответствии с теорией, вероятность того, что значение случайной величины окажется в определенном диапазоне, равна интегралу плотности распределения случайной величины от нижней до верхней границ данного диапазона. Таким образом, может быть вычислено вероятностное значение задержки пакета.

Наиболее часто для описания характеристик производительности сети используются следующие статистические оценки.

**Среднее значение** ( $D$ ) вычисляется как сумма всех значений оцениваемой величины  $d_i$  деленная на количество всех измерений  $N$ :

$$D = \sum \frac{d_i}{N}$$

Для примера, приведенного на рис. 5.7, среднее значение равно:  $(100 \times 25 + 200 \times 35 + 300 \times 45 + 300 \times 55 + 400 \times 65 + 800 \times 75 + 500 \times 85) / 2600 = 64,6$  мс (для вычисления использованы средние значения интервалов).

**Медиана** представляет такое значение оцениваемой величины, которое делит ранжированную (упорядоченную) выборку пополам, то есть таким образом, чтобы количество замеров, значения которых меньше или равны значению медианы, равнялось количеству замеров, значения которых больше или равны значению медианы. В нашем примере медианой выборки является значение 70 мс, так как число замеров, значения которых меньше или равны 70 мс, составляет 1300, а число замеров, значения которых больше или равны 70 мс, равно 1300.

**Стандартное отклонение** ( $J$ ) представляет собой среднее отклонение каждого отдельного замера от среднего значения оцениваемой величины:

$$J = \sqrt{\frac{\sum (d_i - D)^2}{N - 1}}$$

Очевидно, что если все задержки  $d_i$ , равны между собой, то вариация отсутствует, что подтверждают приведенные формулы — в этом случае

$$D = d_i, J = 0.$$

**Коэффициент вариации** — это безразмерная величина, которая равна отношению стандартного отклонения к среднему значению оцениваемой величины:

$$CV = \frac{J}{D}$$

Коэффициент вариации характеризует оцениваемую величину без привязки к ее абсолютным значениям. Так, идеальный равномерный поток пакетов всегда будет обладать нулевым значением коэффициента вариации задержки пакета. Коэффициент вариации задержки пакета, равный 1, означает достаточно пульсирующий трафик, так как средние отклонения интервалов от некоторого среднего периода следования пакетов равны этому периоду.

**Квантиль** (процентиль) — это такое значение оцениваемой величины, которое делит ранжированную выборку на две части так, что процент замеров, значения которых меньше или равно значению квантиля, равен некоторому заданному уровню. В этом определении фигурируют два числа: заранее заданный процент и найденное по нему и замерам выборки значение квантиля. Рассмотрим для примера выборку задержек пакетов, показанную на рис. 5.7, и найдем для нее значение 80-процентного квантиля.

Ответом будет 80 мс, так как ровно 80 % замеров выборки (то есть 2100 замеров из всех интервалов кроме последнего) имеют значения, меньшие или равные 80 мс. Медиана является частным случаем квантиля — это 50-процентный квантиль. Для оценки характеристик сети обычно используют квантили с достаточно большим значением процента, например 90-, 95- или 99-процентные квантили. Это понятно, так как если пользователю скажут, что сеть будет обеспечивать уровень задержек в 100 мс с вероятностью 0,5, то это его не очень обрадует, так как он ничего не будет знать об уровне задержек половины своих пакетов.

## ЗАДАНИЕ

1. Построить гистограмму распределения, найти среднее значение, медиану, стандартное отклонение, коэффициент вариации и квантиль, для заданного диапазона задержек и количества отправленных пакетов.

1) Диапазон задержек 10-90

Отправленные пакеты:  $n_1 = 200$ ,

$n_2 = 250, n_3 = 300, n_4 = 500, n_5 = 700, n_6 = 700, n_7 = 600, n_8 = 200$ .

2) Диапазон задержек 25-105

Отправленные пакеты:  $n_1 = 100$ ,

$n_2 = 300, n_3 = 500, n_4 = 500, n_5 = 650, n_6 = 700, n_7 = 600, n_8 = 400$ .

3) Диапазон задержек 8-72

Отправленные пакеты:  $n_1 = 400$ ,

$n_2 = 500, n_3 = 650, n_4 = 750, n_5 = 800, n_6 = 950, n_7 = 600, n_8 = 300$ .

4) Диапазон задержек 9-81

Отправленные пакеты:  $n_1 = 300$ ,

$n_2 = 500, n_3 = 750, n_4 = 800, n_5 = 700, n_6 = 600, n_7 = 500, n_8 = 300$ .

5) Диапазон задержек 11-99

Отправленные пакеты:  $n_1 = 100$ ,

$n_2 = 250, n_3 = 500, n_4 = 600, n_5 = 350, n_6 = 200, n_7 = 100, n_8 = 100$ .

6) Диапазон задержек 6-54

Отправленные пакеты:  $n_1 = 100$ ,

$n_2 = 200, n_3 = 300, n_4 = 400, n_5 = 500, n_6 = 400, n_7 = 300, n_8 = 200$ .

7) Диапазон задержек 12-108

Отправленные пакеты:  $n_1 = 300$ ,

$n_2 = 450, n_3 = 700, n_4 = 650, n_5 = 400, n_6 = 300, n_7 = 200, n_8 = 200$ .

8) Диапазон задержек 15-135

Отправленные пакеты:  $n_1 = 200$ ,

$n_2 = 200, n_3 = 400, n_4 = 400, n_5 = 600, n_6 = 600, n_7 = 500, n_8 = 500$ .

9) Диапазон задержек 10-90

Отправленные пакеты:  $n_1 = 300$ ,  
 $n_2 = 500, n_3 = 700, n_4 = 600, n_5 = 500, n_6 = 400, n_7 = 300, n_8 = 200$ .

10) Диапазон задержек 5-45

Отправленные пакеты:  $n_1 = 100$ ,  
 $n_2 = 150, n_3 = 200, n_4 = 250, n_5 = 300, n_6 = 450, n_7 = 400, n_8 = 300$ .

2. Написать программу для построения гистограммы распределения, нахождения среднего значения, медиану, стандартного отклонения, коэффициента вариации и квантиля, для заданий 1-10.

#### Содержание отчета

1. Гистограмма распределения, среднее значение, медиана, стандартное отклонение, коэффициент вариации и квантиль для заданного диапазона задержек и количества отправленных пакетов в соответствии с заданием 1.
2. Листинг программы в соответствии с заданием 2.
3. Примеры результатов работы программы.

## ЛИТЕРАТУРА

Издания из фондов библиотеки ВГПУ обозначены \*

### *Основная литература*

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации / В. Л. Бройдо, О. П. Ильина . – 4-е изд. – СПб. : Питер, 2011. – 560 с. : ил.\*
2. Пятибратов А.П. Вычислительные системы, сети и телекоммуникации : учебник / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; под ред. А. П. Пятибратова. – 3-е изд., перераб. и доп. – М. : Финансы и статистика, 2006. – 560 с. : ил.\*

3. Кубряков Е.А. Элементы теории информации и ее представление в памяти компьютера. – Воронеж : Изд-во ВГПУ, 2009. – 71 с.\*
4. Кубряков Е. А. Элементы теории информации и ее представление в памяти компьютера [Электронный ресурс] – Воронеж: Изд-во ВГПУ, 2009. – 71 с. –  
[<URL:http://www.vspu.ac.ru/download/lib/K/K3\\_2009\\_9.pdf>.\\*](http://www.vspu.ac.ru/download/lib/K/K3_2009_9.pdf)
5. Могилев А.В. Средства информатизации. Телекоммуникационные технологии/А. В. Могилев, Л. В. Листрова. – СПб.: БХВ-Петербург, 2009. – 256 с.: ил.\*

### ***Дополнительная литература***

1. Олифер В.Г. Компьютерные сети: принципы, технологии, протоколы/ В.Г.Олифер, Н.А.Олифер. – 3-е изд. – СПб.: Питер, 2006. – 958с. : ил.\*
2. Закер К. Компьютерные сети. Модернизация и поиск неисправностей: Пер. с англ. – СПб.: БХВ-Петербург, 2004.
3. Чулюков В.А. Арифметические основы автоматики и электронно-вычислительной техники: Метод.указания. – Воронеж, ВГПИ, 1987. – 22с.\*
4. Чулюков В.А. Логические элементы и типовые узлы автоматики и электронно-вычислительной техники: Метод.указания. – Воронеж, ВГПИ, 1987. – 25с.\*
5. Чулюков В.А. Основы электронно-вычислительной техники. Центральный процессор: Метод. указания к лаборатор. работам. – Воронеж, ВГПИ, 1989. – 36с.\*
6. Чулюков В.А. Основы электронно-вычислительной техники. Организация памяти: Метод. указания к лаб. работам. – Воронеж, 1989. – 33с.\*
7. Лабораторный практикум по основам функционирования ЭВМ: Учеб.пособие/ Сост.: В.А.Чулюков, М.И.Щевелев и др. – Воронеж, ВГПИ, 1986. – 48с.\*

## Приложение

Приложение 1

Таблица регистрации результатов выполнения директив управления моделью ЭВМ

Директива	АК	РВВ	СЧАК	РК			РБА	РА	РС
				КОП	ТА	А			

Приложение 2

Таблица регистрации результатов выполнения директив программы в режиме ЦИКЛ

СЧАК	РК			РО	АК	Примечание
	КОП	ТА	А			

Приложение 3

Таблица регистрации результатов выполнения программы в режиме ТАКТ

№ такта	СЧАК	РК			№МК	РА	РС	РО	АК	Примечание
		КОП	ТА	А						

## **Оглавление**

Введение .....	3
1. Архитектура ЭВМ с минимальной системой команд .....	4
1.1. Структура ЭВМ .....	4
1.2. Представление данных команд .....	3
1.3. Состояния и режимы работы ЭВМ .....	5
1.4. Взаимодействие пользователя с моделью ЭВМ. ....	7
2. Изучение архитектуры ЭВМ.....	12
2.1. Лабораторная работа №1. Ознакомление с архитектурой ЭВМ, директивами управления моделью ЭВМ .....	12
2.2. Лабораторная работа №2. Программирование алгоритма без цикла .....	17
2.3. Лабораторная работа №3. Программирование цикла с переадресацией .....	25
3. Изучение алгоритма работы центрального устройства управления ЭВМ.....	29
3.1. Принцип микропрограммного управления.....	29
3.2. Микропрограмма центрального устройства управления ЭВМ .....	31
3.3. Работа модели ЭВМ в режиме ТАКТ .....	35
3.4. Средства контроля знаний студентов .....	35
3.5. Лабораторная работа №4. Изучение основного машинного цикла .....	37
3.6. Лабораторная работа №5. Изучение микропрограммы арифметических операций.....	39
3.7. Лабораторная работа №6. Изучение микропрограммы посыльных операций .....	42
3.8. Лабораторная работа №7. Изучение микропрограммы операций передачи управления .....	44
4. Управление виртуальной ЭВМ.....	47
4.1. Директивы управления системой .....	47
4.2. Параметры модели .....	47
4.3. Кодирование неисправностей .....	48
5. Сети коммуникаций .....	50

<i>5.1. Маршрутизация</i> .....	50
<i>5.2. Лабораторная работа №8. Поиск оптимального маршрута по критерию пропускной способности коммуникационной сети</i> .....	52
<i>5.3. Лабораторная работа №9. Методы формирования плана распределения информации</i> .....	57
<i>5.4. Лабораторная работа №10. Статистические оценки характеристик сети</i> .	64
<b>ЛИТЕРАТУРА</b> .....	69
<i>Основная литература</i> .....	69
<i>Дополнительная литература</i> .....	70
<i>Приложение</i> .....	71