

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 15.06.2023 10:11:51

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

**МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования**

**«Юго-Западный государственный университет»  
(ЮЗГУ)**

**Кафедра информационных систем и технологий**

**УТВЕРЖДАЮ**  
Проректор по учебной работе  
**О.Г. Локтионова**  
« 4 » 02 (ЮЗГУ) 2018 г.



**Вычислительные системы, сети и телекоммуникации**

Методические указания к лабораторному  
практикуму по дисциплине «Вычислительные  
системы, сети и телекоммуникации» для  
студентов, обучающихся по направлению 09.03.03  
«Прикладная информатика»

Курск 2018

УДК 004  
Составитель И.В. Зотов

Рецензент  
Кандидат технических наук, профессор Шлыков В.А.

**Вычислительные системы, сети и телекоммуникации:**  
**Методические указания к лабораторному практикуму / Юго-**  
Зап. гос. ун-т; сост. И.В. Зотов. Курск, 2018. 57 с. Библиогр.: с. 56.

Содержат сведения по основам вычислительных систем, построения сетей и средствам телекоммуникаций. Указывается порядок выполнения практических работ, содержание отчета и правила оформления практических работ, контрольные вопросы по теме работы.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям информатики и вычислительной техники (УМО ИиВТ).

Предназначены для студентов направления подготовки 09.03.03 дневной и заочной формы обучения.

Текст печатается в авторской редакции.

Подписано в печать *1.02*. Формат 60x84 1/16.  
Усл.печ. л. 3,31. Уч.-изд. л. 2,9. Тираж *100* экз. Заказ. *325*  
Бесплатно.

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## Практическая работа №1

### Системы счисления

#### *Теоретический материал.*

Под **системой счисления** понимается способ представления любого числа с помощью некоторого алфавита символов, называемых цифрами.

Все системы счисления делятся на **позиционные** и **непозиционные**.

**Непозиционными** системами являются такие системы счисления, в которых каждый символ сохраняет свое значение независимо от места его положения в числе.

Примером непозиционной системы счисления является римская система. К недостаткам таких систем относятся наличие большого количества знаков и сложность выполнения арифметических операций.

Система счисления называется **позиционной**, если одна и та же цифра имеет различное значение, определяющееся позицией цифры в последовательности цифр, изображающей число. Это значение меняется в однозначной зависимости от позиции, занимаемой цифрой, по некоторому закону.

Примером позиционной системы счисления является десятичная система, используемая в повседневной жизни.

Количество  $p$  различных цифр, употребляемых в позиционной системе определяет название системы счисления и называется **основанием** системы счисления - " $p$ ".

В десятичной системе используются десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; эта система имеет основанием число десять.

Любое число  $N$  в позиционной системе счисления с основанием  $p$  может быть представлено в виде полинома от основания  $p$ :

$$N = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0 + a_{-1} p^{-1} + a_{-2} p^{-2} + \dots$$

здесь  $N$  - число,  $a_j$  - коэффициенты (цифры числа),  $p$  - основание системы счисления ( $p > 1$ ).

Принято представлять числа в виде последовательности цифр:

$$N = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots$$

В этой последовательности точка отделяет целую часть числа

от дробной (коэффициенты при положительных степенях, включая нуль, от коэффициентов при отрицательных степенях). Точка опускается, если нет отрицательных степеней (число целое).

Таблица 1. Наиболее важные системы счисления

Двоичная (Основание 2)	Восьмеричная (Основание 8)		Десятичная (Основание 10)	Шестнадцатеричная (Основание 16)	
		триады			тетрады
0	0	000	0	0	0000
1	1	001	1	1	0001
	2	010	2	2	0010
	3	011	3	3	0011
	4	100	4	4	0100
	5	101	5	5	0101
	6	110	6	6	0110
	7	111	7	7	0111
			8	8	1000
			9	9	1001
				A	1010
				B	1011
				C	1100
				D	1101
				E	1110
				F	1111

В ЭВМ применяют позиционные системы счисления с недесятичным основанием: двоичную, восьмеричную, шестнадцатеричную.

В аппаратной основе ЭВМ лежат двухпозиционные элементы, которые могут находиться только в двух состояниях; одно из них обозначается 0, а другое - 1. Поэтому основной системой счисления применяемой в ЭВМ является двоичная система.

**Двоичная система счисления.** Используется две цифры: 0 и 1. В двоичной системе любое число может быть представлено в виде:

$$N = b_n b_{n-1} \dots b_1 b_0 . b_{-1} b_{-2} \dots$$

где  $b_j$  либо 0, либо 1.

**Восьмеричная система счисления.** Используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Употребляется в ЭВМ как вспомогательная для записи информации в сокращенном виде. Для представления одной цифры восьмеричной системы используется три двоичных разряда (триада) (Таблица 1).

**Шестнадцатеричная система счисления.** Для изображения чисел употребляются 16 цифр. Первые десять цифр этой системы обозначаются цифрами от 0 до 9, а старшие шесть цифр - латинскими буквами: 10-A, 11-B, 12-C, 13-D, 14-E, 15-F. Шестнадцатеричная система используется для записи информации в сокращенном виде. Для представления одной цифры шестнадцатеричной системы счисления используется четыре двоичных разряда (тетрада) (Таблица 1).

**Рассмотрим перевод чисел из одной системы счисления в другую.**

**Перевод чисел в десятичную систему** осуществляется путем составления степенного ряда с основанием той системы, из которой число переводится. Затем подсчитывается значение суммы.

Пример.

а) Перевести  $10101101.101_2 \rightarrow "10"$  с.с.

$$10101101.101_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 133.125_{10}$$

б) Перевести  $703.04_8 \rightarrow "10"$  с.с.

$$703.04_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 3 \cdot 8^0 + 0 \cdot 8^{-1} + 4 \cdot 8^{-2} =$$

$451.0625_{10}$  в) Перевести  $B2E.4_{16} \rightarrow "10"$  с.с.

$$B2E.4_{16} = 11 \cdot 16^2 + 2 \cdot 16^1 + 14 \cdot 16^0 + 4 \cdot 16^{-1} = 2862.25_{10}$$

**Перевод целых десятичных чисел в недесятичную систему счисления** осуществляется последовательным делением десятичного числа на основание той системы, в которую оно

переводится, до тех пор, пока не получится частное меньше этого основания. Число в новой системе записывается в виде остатков деления, начиная с последнего.

Пример.

а) Перевести  $181_{10} \rightarrow "8"$  с.с.

$$\begin{array}{r|l} 181 & 8 \\ -176 & 22 \\ \hline 5 & 16 \\ & 2 \\ \hline & 6 \end{array}$$

Результат:  $181_{10} = 265_8$

б) Перевести  $622_{10} \rightarrow "16"$  с.с.

$$\begin{array}{r|l} 622 & 16 \\ -48 & 38 \\ \hline 142 & 32 \\ -128 & 6 \\ \hline 14 & \end{array}$$

Результат:  $622_{10} = 26E_{16}$

**Перевод правильных дробей из десятичной системы счисления в недесятичную.**

Для перевода правильной десятичной дроби в другую систему эту дробь надо последовательно умножать на основание той системы, в которую она переводится. При этом умножаются только дробные части. Дробь в новой системе записывается в виде целых частей произведений, начиная с первого.

Пример.

Перевести  $0.3125_{10} \rightarrow "8"$  с.с.

$$\begin{array}{l} \downarrow \\ \begin{array}{r|l} 0 & 3125 \times 8 \\ \hline 2 & 5000 \times 8 \\ 4 & 0000 \end{array} \end{array}$$

Результат:  $0.3125_{10} = 0.24_8$

**Замечание.** Конечной десятичной дроби в другой системе счисления может соответствовать бесконечная (иногда периодическая) дробь. В этом случае количество знаков в представлении дроби в новой системе берется в зависимости от требуемой точности.

Пример.

Перевести  $0.65_{10} \rightarrow "2"$  с.с. Точность 6 знаков.

0	65 × 2
1	3 × 2
0	6 × 2
1	2 × 2
0	4 × 2
0	8 × 2
1	6 × 2
...	

Результат:  $0.65_{10} \approx 0.10(1001)_2$

**Для перевода неправильной десятичной дроби в систему счисления с недесятичным основанием необходимо отдельно перевести целую часть и отдельно дробную.**

Пример:

Перевести  $23.125_{10} \rightarrow$  "2" с.с.

1) Переведем  
целую часть:

23	2
22	11 2
1	10 5 2
	1 4 2 2
	1 2 1
	0

2) Переведем  
дробную часть:

0	125 × 2
0	25 × 2
0	5 × 2
1	0

Таким образом:  $23_{10} = 10111_2$ ;  $0.125_{10} = 0.001_2$ .

Результат:  $23.125_{10} = 10111.001_2$ .

Необходимо отметить, что целые числа остаются целыми, а правильные дроби - дробями в любой системе счисления.

**Для перевода восьмеричного или шестнадцатеричного числа в двоичную форму достаточно заменить каждую цифру этого числа соответствующим трехразрядным двоичным числом (триадой) (таблица 1) или четырехразрядным двоичным числом (тетрадой) (таблица 1), при этом отбрасывают ненужные нули в старших и младших разрядах.**

Пример.

а) Перевести  $305.4_8 \rightarrow$  "2" с.с.

$$\begin{array}{cccc} \underline{3} & \underline{0} & \underline{5} & \underline{.4} \\ \underline{011} & \underline{000} & \underline{101} & \underline{100} \end{array} = 11000101.1_2$$

б) Перевести  $7B2.E_{16} \rightarrow "2" \text{ с.с.}$

$$\begin{array}{cccc} \underline{7} & \underline{B} & \underline{2} & \underline{.E} \\ \underline{0111} & \underline{1011} & \underline{0010} & \underline{1110} \end{array} = 11110110010.111_2$$

**Для перехода от двоичной к восьмеричной (шестнадцатеричной) системе** поступают следующим образом: двигаясь от точки влево и вправо, разбивают двоичное число на группы по три (четыре) разряда, дополняя при необходимости нулями крайние левую и правую группы. Затем триаду (тетраду) заменяют соответствующей восьмеричной (шестнадцатеричной) цифрой.

Пример.

а) Перевести  $1101111001.1101_2 \rightarrow "8" \text{ с.с.}$

$$\begin{array}{cccccc} \underline{001} & \underline{101} & \underline{111} & \underline{001} & \underline{.110} & \underline{100} \\ \underline{1} & \underline{5} & \underline{7} & \underline{1} & \underline{6} & \underline{4} \end{array} = 1571.64_8$$

б) Перевести  $1111111011.100111_2 \rightarrow "16" \text{ с.с.}$

$$\begin{array}{cccccc} \underline{0111} & \underline{1111} & \underline{1011} & \underline{.1001} & \underline{1100} \\ \underline{7} & \underline{F} & \underline{B} & \underline{9} & \underline{C} \end{array} = 7FB.9C_{16}$$

**Перевод из восьмеричной в шестнадцатеричную систему и обратно** осуществляется через двоичную систему с помощью триад и тетрад.

Пример. Перевести  $175.24_8 \rightarrow "16" \text{ с.с.}$

$$\begin{array}{cccccc} \underline{1} & \underline{7} & \underline{5} & \underline{.2} & \underline{4} \\ \underline{001} & \underline{111} & \underline{101} & \underline{010} & \underline{100} \end{array} = 1111101.0101_2 = \begin{array}{ccc} \underline{0111} & \underline{1101} & \underline{.0101} \\ \underline{7} & \underline{D} & \underline{5} \end{array} = 7D.5_{16}$$

Результат:  $175.24_8 = 7D.5_{16}$ .

## 1. Двоичная арифметика.

Правила выполнения арифметических действий над двоичными числами задаются таблицами двоичных сложения, вычитания и умножения.

При сложении двоичных чисел в каждом разряде производится сложение цифр слагаемых и переноса из соседнего младшего разряда, если он имеется. При этом необходимо учитывать, что  $1+1$  дают нуль в данном разряде и единицу переноса в следующий.



Таблица 2. Правила выполнения арифметических действий над двоичными числами

Таблица двоичного сложения	Таблица двоичного вычитания	Таблица двоичного умножения
$0+0=0$	$0-0=0$	$0 \times 0=0$
$0+1=1$	$1-0=1$	$0 \times 1=0$
$1+0=1$	$1-1=0$	$1 \times 0=0$
$1+1=10$	$10-1=1$	$1 \times 1=1$

Пример. Выполнить сложение двоичных чисел:  
а)  $X=1101$ ,  $Y=101$ ;

$$\begin{array}{r}
 \phantom{X=} \phantom{Y=} \phantom{X+Y=} \phantom{1} \phantom{1} \phantom{\leftarrow} \text{единицы переноса} \\
 \phantom{X=} \phantom{Y=} \phantom{X+Y=} \phantom{1} \phantom{1} \phantom{\leftarrow} \\
 X= \phantom{+} 1101 \\
 Y= + \phantom{1} 101 \\
 \hline
 X+Y= 10010
 \end{array}$$

Результат  $1101+101=10010$ .

б)  $X=1101$ ,  $Y=101$ ,  $Z=111$ ;

$$\begin{array}{r}
 \phantom{X=} \phantom{Y=} \phantom{Z=} \phantom{X+Y+Z=} \phantom{1} \phantom{\leftarrow} \text{единицы переноса} \\
 \phantom{X=} \phantom{Y=} \phantom{Z=} \phantom{X+Y+Z=} \phantom{1} \phantom{\leftarrow} \\
 X= \phantom{+} 1101 \\
 Y= + \phantom{1} 101 \\
 Z= + \phantom{1} 111 \\
 \hline
 X+Y+Z= 11001
 \end{array}$$

Результат  $1101+101+111=11001$ .

При вычитании двоичных чисел в данном разряде при необходимости занимается 1 из старшего разряда. Эта занимаемая 1 равна двум 1 данного разряда.

Пример. Заданы двоичные числа  $X=10010$  и  $Y=101$ .  
Вычислить  $X-Y$ .

$$\begin{array}{r}
 10010 \\
 -101 \\
 \hline
 01101
 \end{array}$$

Результат  $10010 - 101=1101$ .

Умножение двоичных чисел производится по тем же

правилам, что и для десятичных с помощью таблиц двоичного умножения и сложения.

Пример.  $1001 \times 101 = ?$

$$\begin{array}{r} 1001 \\ \times 101 \\ \hline 1001 \\ 1001 \\ \hline 101101 \end{array}$$

Результат  $1001 \times 101 = 101101$ .

Деление двоичных чисел производится по тем же правилам, что и для десятичных. При этом используются таблицы двоичного умножения и вычитания.

Пример.  $1100.011 : 10.01 = ?$

$$\begin{array}{r} 110001.1 \quad | \quad 1001 \\ \underline{1001} \quad | \quad 101.1 \\ 1101 \\ \underline{1001} \\ 1001 \\ \underline{1001} \\ 0 \end{array}$$

Результат  $1100.011 : 10.01 = 101.1$ .

### Варианты задания

1. Перевести следующие числа в десятичную систему счисления:

а)  $110111_2$ ; б)  $10110111.1011_2$ ; в)  $563.44_8$ ; г)  $721.35_8$ ; д)  $1C4.A_{16}$ ; е)  $9A2F.B5_2$ .

2. Перевести следующие числа из "10" с.с в "2", "8", "16" с.с.:

а) 463; б) 1209; в) 362; г) 3925; д) 11355.

3. Перевести следующие числа из "10" с.с в "2", "8", "16" с.с. (точность вычислений - 5 знаков после точки):

а) 0.0625; б) 0.345; в) 0.225; г) 0.725; д) 217.375; е) 31.2375; ж) 725.03125; з) 8846.04.

4. Перевести следующие числа в двоичную систему счисления:

а)  $1725.326_8$ ; б)  $341.34_8$ ; в)  $7BF.52A_{16}$ ; г)  $3D2.C_{16}$ .

5. Перевести следующие числа из одной системы счисления в

другую:

- а)  $11011001.01011_2 \rightarrow "8" \text{ с.с.};$
- б)  $1011110.1101_2 \rightarrow "8" \text{ с.с.};$
- в)  $1101111101.0101101_2 \rightarrow "16" \text{ с.с.};$
- г)  $110101000.100101_2 \rightarrow "16" \text{ с.с.}$

6. Перевести следующие числа из одной системы счисления в другую:

- а)  $312.7_8 \rightarrow "16" \text{ с.с.};$     б)  $51.43_8 \rightarrow "16" \text{ с.с.};$
- в)  $5B.F_{16} \rightarrow "8" \text{ с.с.};$     г)  $D4.19_{16} \rightarrow "8" \text{ с.с.}$

7. Заданы двоичные числа  $X$  и  $Y$ . Вычислить  $X+Y$  и  $X-Y$ , если:

- а)  $X=1101001; Y=101111;$
- б)  $X=101110110; Y=10111001;$
- в)  $X=100011001; Y=101011.$

8. Заданы двоичные числа  $X$  и  $Y$ . Вычислить  $X*Y$  и  $X/Y$ , если:

- а)  $X=1000010011; Y=1011;$
- б)  $X=110010101; Y=1001;$
- в)  $X=100101.011; Y=110.1;$
- г)  $X=100000.1101; Y=101.01.$

## Практическая работа №2

### Извлечение команд

#### *Теоретический материал.*

Микропроцессор (МП) - основная полупроводниковая схема, выполняющая в компьютере обработку информации и содержащая арифметико-логическое устройство, регистровую память, КЭШ, устройство управления, шину сопряжения компонентов, порты (адресуемые регистры ввода-вывода) и т. д. КЭШ - быстродействующая память относительно небольшого объема, предназначенная для временного хранения часто используемых данных.

Микропроцессор характеризуется набором команд, системой адресации команд и данных, типом (форматом) обрабатываемых данных, конвейерной обработкой и другими показателями.

Операционное устройство - программно-управляемое устройство обработки информации, являющееся частью оборудования МП, и содержащее арифметико-логическое устройство и регистровую память, выполняющее список микроопераций (синхросигналов), необходимых для реализации различных команд (арифметических, условных переходов, пересылки данных, сдвига, логических и других). В слове состояния операционного устройства устанавливаются в конце выполнения команды соответствующие флаги (биты-признаки), например, старший бит знака, переноса, паритета, нуля или переполнения. Память - оперативное запоминающее устройство, хранящее загружаемую операционную систему команды и данные программ, обрабатываемые микропроцессором в текущее время, в которую операционная система помещает прикладные программы, считываемые с диска для их выполнения. Память характеризуется объемом и быстродействием. Различают память статического и динамического способа действия. Память динамического типа требует периодической регенерации (восстановления) информации, содержащейся в последовательно расположенных ячейках памяти.

Устройство управления - устройство, координирующее работу компьютера и обеспечивающее управление операциями в нем

путем формирования микроопераций (синхросигналов), генерируемых в соответствии с кодом операции выполняемой команды. Наиболее широкое применение получили устройства с "хранимой логикой", которые содержат микропрограммную память (ПИЗУ), предназначенную для хранения последовательностей микрокоманд (элементарных операций), обеспечивающих выполнение машинных команд.

Счётчик команд - схема, предназначенная для временного хранения адреса команды, размещенной в памяти, содержимое которой, увеличивается для реализации последовательной (естественной) адресации ячеек памяти, и для хранения адреса перехода в случае необходимости нарушения последовательной адресации.

Регистр команд - регистр, служащий для временного хранения исполняемой команды.

Дешифратор команд (кода операций) - схема, определяющая последовательность формирования синхросигналов (микроопераций) устройство управления для реализации текущей команды программы.

### *Выполнение работы*

Интерфейс разработанного приложения представлен на рис 1.

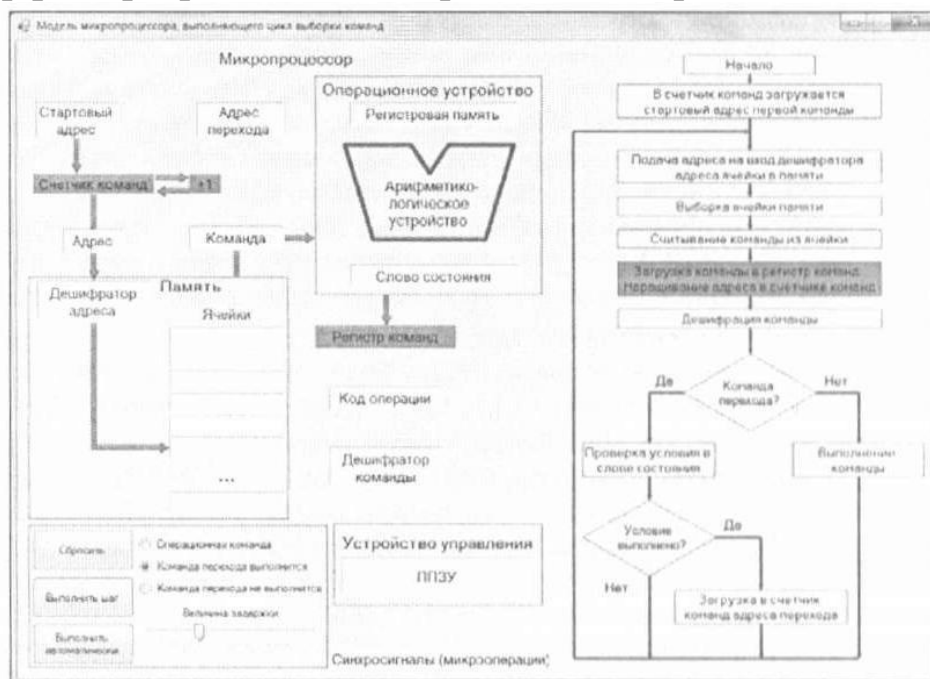


Рис. 1. Окно программы

Предлагаемая демонстрационная модель микропроцессора, подробно показывает процесс циклической выборки команд. Выполнение каждой итерации процесса выборки команд сопровождается подсветкой соответствующих вычислительных блоков микропроцессора и вершин на блок-схеме алгоритма работы.

Модель реализует цикл выборки команд и позволяет продемонстрировать работу микропроцессора и следующих случаях:

1. Выбрана операционная команда.
2. Выбрана команда перехода, но переход не будет осуществлен (не выполнены условия перехода).
3. Выбрана команда перехода, и переход будет выполнен.

В левой нижней части приложения расположена панель, на которой находятся все управляющие элементы, это:

1. Кнопки запуска моделирования и сброса в начальное состояние.
2. Группа кнопок `RadioButton`, выбора типа команды.
3. Ползунок для регулирования задержки в автоматическом режиме.

Запустить программу можно в двух режимах (соответственно по названию кнопок):

а) пошаговый: пользователь управляет переключением модели на  $j$  следующий шаг алгоритма;

б) автоматический: программа продемонстрирует процесс моделирования. Задержка между шагами регулируется пользователем с помощью ползунка.

На каждом шаге моделирования приложение подсвечивает выполняемый блок в блок-схеме моделируемого алгоритма (расположен в правой части приложения) и блоки устройств в модели, которые задействованы в текущем шаге, или блоки данных, пересылаемых другому блоку модели.

Выбрать тип команды, обработку которой демонстрирует модель на текущем проходе цикла выборки, пользователь может с помощью группы кнопок `RadioButton`, модель подстроится подтип команды в процессе моделирования.

Для получения дополнительной информации о блоке процессора предусмотрена функция всплывающей подсказки, пользователь может кликнуть по одному из блоков процессора и получить сообщение с подробным описанием его состава и функций, которые он выполняет. Модель цикла выборки команд в микропроцессоре представлена на диске в папке «Извлечение команд».

## Практическая работа №3

### Конвейерная обработка

#### *Теоретический материал.*

Выполнение команды микропроцессора (МП) содержит несколько этапов:

1. Выборка команды из памяти.
2. Дешифрация текущей команды.
3. Чтение операндов из памяти, если это требуется.
4. Реализация команды.
5. Запись результата в память, если это действие задано в команде.

Шаги последовательного выполнения команды, использующей внутренние регистры общего назначения, представлены на рис. 2.

Адрес	Выборка	Дешифрация	Выполнение
-------	---------	------------	------------

Рис. 2. Шаги выполнения простой команды

В том случае, когда результат выполнения необходимо записать в память, эти шаги выглядят следующим образом (рис. 3).

Адрес	Выборка	Дешифрация	Выполнение	Адрес	Запись
-------	---------	------------	------------	-------	--------

Рис. 3. Шаги выполнения команды с записью результата в память

Для выполнения этой команды требуется шесть шагов. Если команды выполнить последовательно друг за другом, то будет наблюдаться ощутимый простой оборудования микропроцессора. Поэтому в нем реализуется параллельное (конвейерное) выполнение сразу нескольких команд. связанное с числом его блоков, путем использования буферов, в которых задаются очереди команд и результатов их дешифрации (рис. 4).



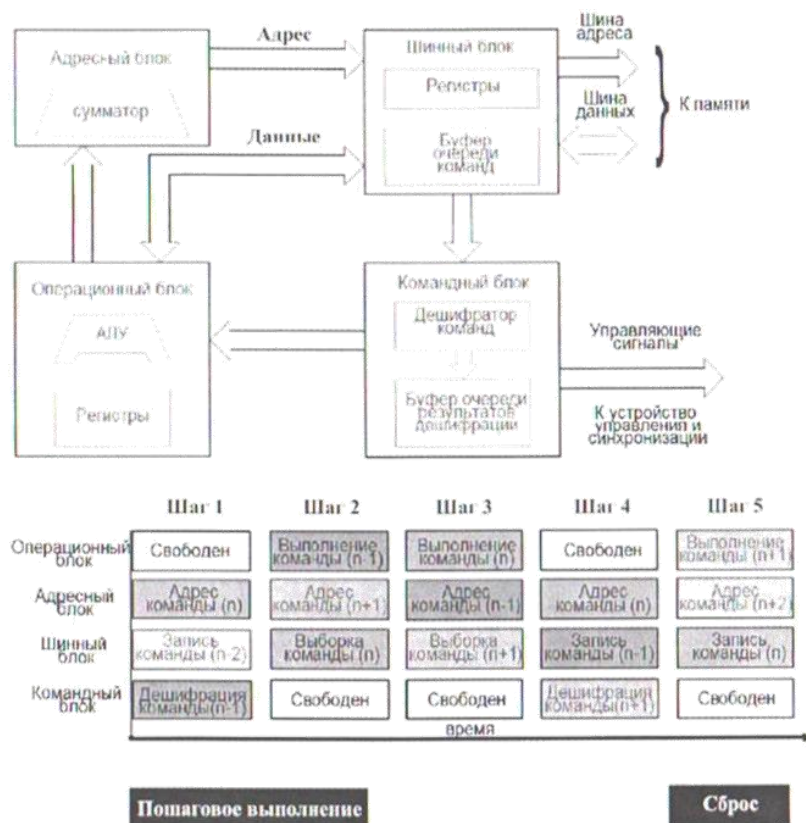


Рис. 4. Модель конвейерного выполнения команд

Микропроцессор (МП) является сложным и универсальным устройством, предназначенным для выполнения различных программ, связанных с обработкой информации. В общем случае МП содержит в упрощенном варианте четыре основных блока: операционный, адресный, шинный, командный.

Центральным компонентом операционного блока является арифметико-логическое устройство (АЛУ) и совокупность регистров специального и общего назначения, с помощью которых выполняются различные операции: сложения, сдвига, логики, сравнения, пересылки данных и т. д.

С увеличением объема памяти возрастает ее адресное пространство и длина адресного слова. Для доступа к ячейкам памяти используется адресный блок, который формирует с помощью сумматора  $n$ -разрядный код адреса. Различают линейную и сегментную адресацию. В первом случае, все адресное пространство рассматривается как единое целое, а во втором, вся память делится на сегменты, каждый из которых имеет свой номер.

Поэтому адрес представляется как помер сегмента и некоторого смещения в нем.

Шинный блок обеспечивает доступ к памяти и организацию шинных операций (записи и чтения данных с реализацией протоколов связи). Он выполняет выборку команд программы из памяти и организует в своем буфере (внутренней памяти) очередь команд, что позволяет операционному блоку сразу выполнять команду из очереди, не ожидая ее считывания из памяти. Командный блок осуществляет декодирование кода операции команды, указывающий операционному блоку микропрограмму управления и синхронизации выполнения заданной операции. Результаты дешифрации временно хранятся в буфере.

#### *Выполнение работы*

Модель демонстрирует пошаговое выполнение фрагмента команд, причем каждая команда помечена соответствующим цветом, и ее выполнение в каждом блоке МП отмечается этим же цветом. Выполнить пошагово выполнение команд на модели. Оценить, на какой блок приходится наибольшая нагрузка, на какой блок - наименьшая, и почему. Для увеличения производительности МП необходимо повысить пропускную способность каких блоков? Модель конвейерного выполнения команд представлена на диске в папке «Конвейерная обработка».

## Практическая работа №4 Трёхшинная конфигурация

### *Теоретический материал.*

Для управления работой компонентов в системе микропроцессор использует шину данных, шину адреса и шину управления (рис. 5).

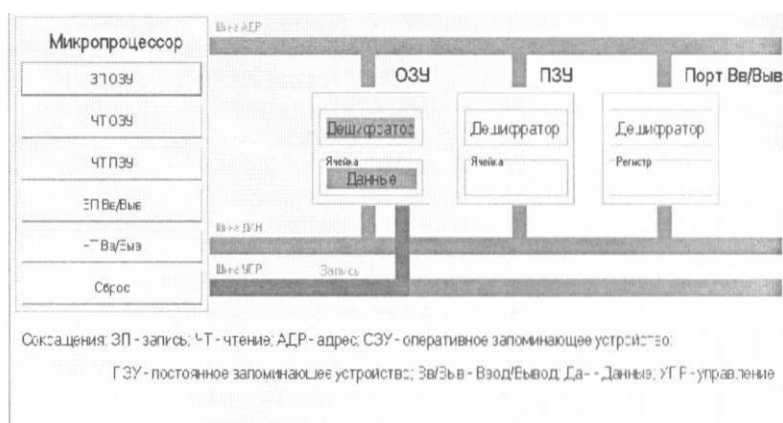


Рис. 5. Трёхшинная конфигурация

Шина данных - это двунаправленная шина с тремя состояниями, к которой в общем случае подключены многочисленные источники и приемники информации, но в текущий момент времени к шине для совместной работы подключены только два устройства: активное (микропроцессор) и пассивное, например, оперативная память (ОЗУ). Третье состояние шины обеспечивается специальными шинными формирователями, характеризующимися высоким сопротивлением (состояние отключено), если их работа не разрешена соответствующим сигналом.

ОЗУ является источником и приемником данных, поэтому микропроцессор (МП) выполняет операцию записи или чтения памяти. В первом случае МП устанавливает адрес на шину адреса для выбора ячейки памяти и данные на шину данных, а затем направляет сигнал записи по соответствующей линии шины управления. В режиме чтения МП также устанавливает адрес на шину адреса и сигнал чтения памяти на одноименной линии шины

управления, который вызывает появление данных на шине данных. После этого МП принимает эти данные в свой внутренний регистр.

Аналогичным образом осуществляется доступ к порта ввода и вывода (Вв/Выв) с учетом операций записи (Зп Вв/Выв) и чтения (Чт Вв/Выв), выполняемых МП. Для считывания данных из ПЗУ МП устанавливает адрес и сигнал чтения ПЗУ, вызывающий появление данных на соответствующий шине.

Для опознавания устройств, применяются дешифраторы адресов, предназначенные для выбора конкретной ячейки памяти или порта ввода/вывода.

При нажатии кнопки, например, запись Зп ОЗУ на экране визуально демонстрируется: передача адреса по соответствующей шине адреса (АДР), его дешифрация, вывод данных по шине данных (ДАН) и установка сигнала записи в ячейку по шине УПР.

Сигнал записи является магистральным, то есть является доступным для всех устройств, но только в одном устройстве, опознавшим собственный адрес, произойдет запись данных в ячейку или регистр.

При чтении данных из ПЗУ также показывается передача адреса по шине адреса (АДР) и установка магистрального сигнала чтение, вызывающего появление данных на шине ДАН. Эти действия происходят по нажатию кнопки Чит ПЗУ, причем в ячейке ПЗУ демонстрируется подсветкой наличие информации. Модель трёхшинной конфигурации показана на диске в папке «Трёхшинная конфигурация».

#### *Выполнение работы.*

Выполнить последовательно операции чтения ОЗУ, запись в ОЗУ, чтение ПЗУ, чтение из порта ввода/вывода, запись в порт ввода/вывода. Описать порядок прохождения каждой операции (приложить соответствующий скриншот выполнения операции на модели).

## Практическая работа №5 Выполнение режимов доступа

### *Теоретический материал.*

Рассматривается модель, которая позволяет наглядно демонстрировать работу четырёх базовых операций: чтения из оперативной памяти, записи в оперативную память, вывод в порт и ввод из порта.

### *Выполнение работы.*

Чтение оперативной памяти (рис. 6 и рис. 7).

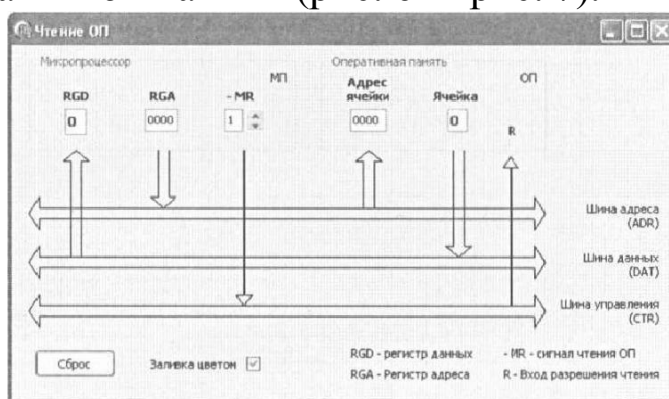


Рис. 6. Состояние сброса

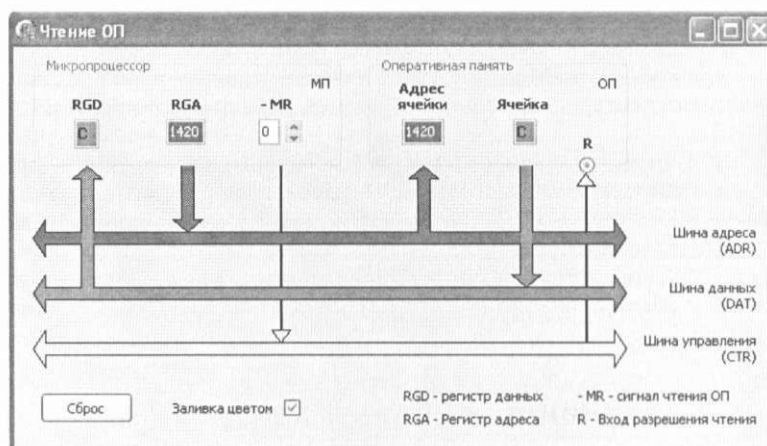


Рис. 7. Чтение ОП (передача данных)

Порядок действий:

- 1) Ввести символ, например, С в ячейку ОП.
- 2) Набрать четырёхразрядный 16-ричный адрес в регистре RGA (1420). При сбросе RGA-(0000).
- 3) Установить активный сигнал чтение - MR = 0. При сбросе - MR = 1.

4) Содержимое ячейки памяти (C) переместить в регистр RGD. При сбросе RGD пустой (0).

Запись в оперативную память (рис. 8 и рис. 9). Порядок действий:

1. Ввести символ, например, F в регистр RGD.
2. Набрать адрес (1560h) в регистр RGA.
3. Установить активный сигнал записи в память - MW = 0.
4. Содержимое RGD (F) переместить в ячейку ОП.

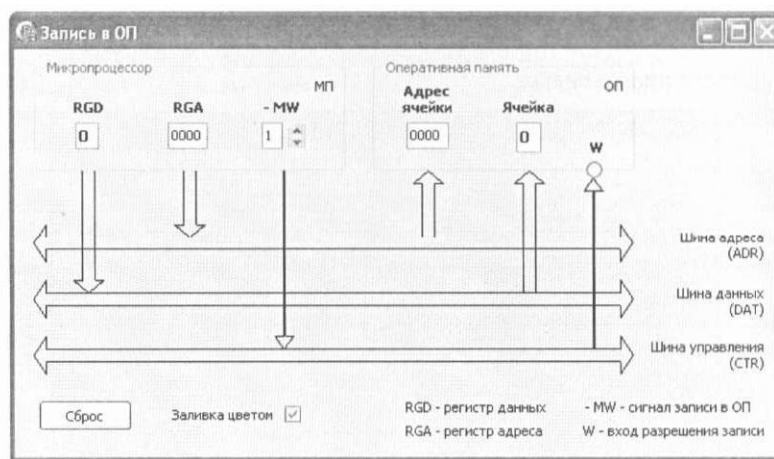


Рис. 8. Состояние сброса

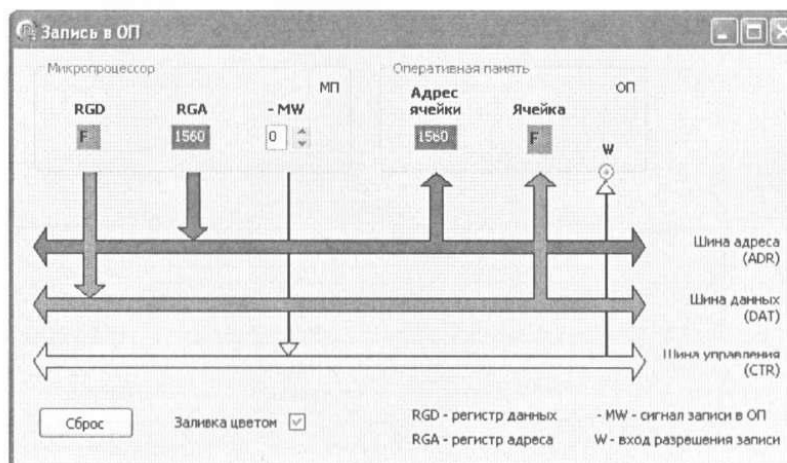


Рис. 9. Запись в ОП (передача данных)

Вывод в порт (рис. 10 и рис. 11). Порядок действий:

1. Ввести какой-либо символ (13) в регистр RGD.
2. Задать адрес 379h в регистр RGA, который передаётся в окно адреса порта.
3. Установить сигнал - WR = 0.

4. При этом в регистре вывода RGO порта появляется символ В.

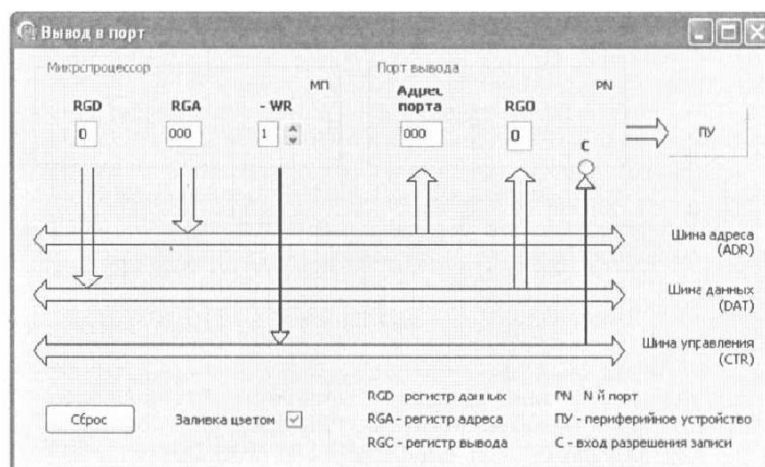


Рис. 10. Состояние сброса

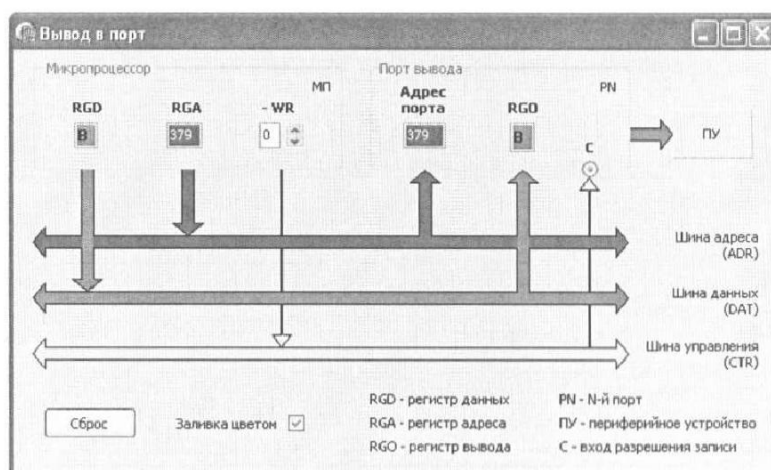


Рис. 11. Вывод в порт (передача данных)

Ввод из порта (рис. 12 и рис. 13). Порядок действий:

- 1) Ввести в регистр ввода RGI какой-нибудь символ, например, В
- 2) Задать адрес 365h в регистр RGA.
- 3) Установить сигнал - RD = 0.
- 4) Переместить содержимое RGI в регистр RGD.

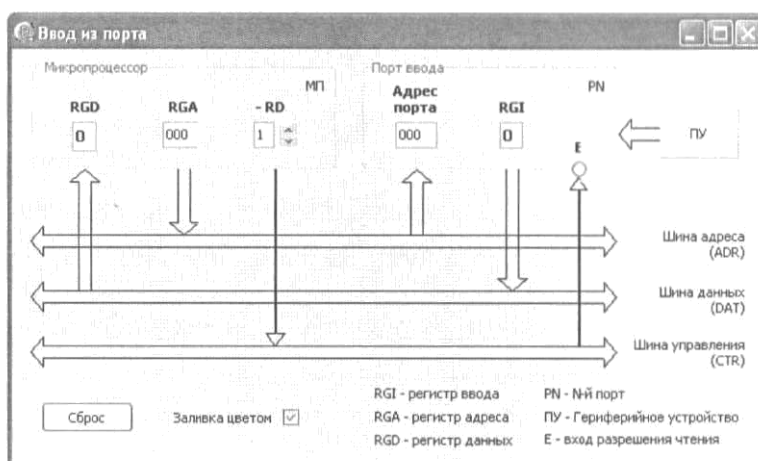


Рис. 12. Состояние сброса

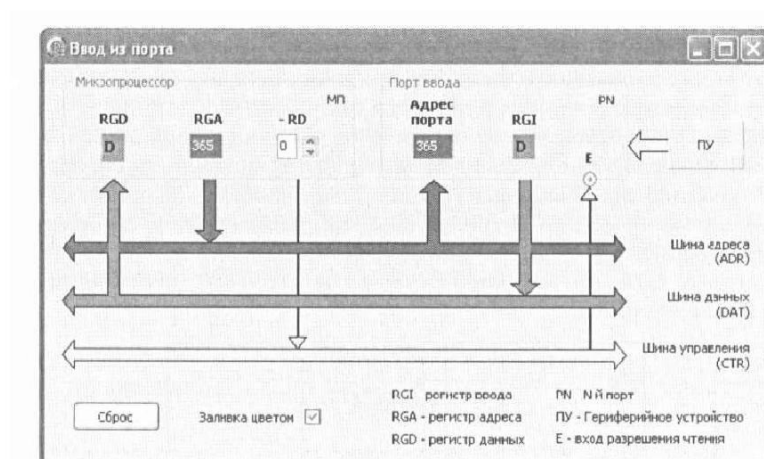


Рис. 13. Ввод из порта (передача данных)

Модель режимов доступа представлена на диске в папке «Выполнение режимов доступа»



## Практическая работа №6 Организация памяти

### *Теоретический материал.*

Память с произвольной выборкой (основная память) - это устройство хранения информации (команд, данных), которая может быть извлечена в любой момент времени для дальнейшей ее обработки. Основная память есть запоминающее устройство (энергозависимое, для считывания и записи текущей информации), поэтому его называют еще оперативным (ОЗУ). Известны несколько типов ОЗУ: статические и динамические. В статических ОЗУ основным элементом памяти является триггер, а в динамических - конденсатор, который требует периодической подзарядки (регенерации) для сохранения единичного бита информации (наличие заряда). В режиме регенерации, каждый раз, через определенное время, выполняется считывание значений из ячеек памяти и автоматическая их перезапись. Там, где хранится логическая 1, присутствует на конденсаторе заряд, а где ее нет - заряд отсутствует, что соответствует логическому 0. Для реализации большой емкости памяти наиболее употребительны ОЗУ динамического типа, так как для построения элемента памяти, хранящего бит информации, требуется меньшее число транзисторов, чем в статической памяти. Кроме того, динамическая память потребляет меньше энергии (мощности). Недостатком ОЗУ динамического типа является наличие дополнительных схем регенерации памяти. Поэтому для создания памяти небольшой емкости используют ОЗУ статического типа. Микросхемы ОЗУ имеют битовую или байтовую (словарную) организацию, например,  $64\text{К} \times 1$  или  $2\text{К} \times 8$ , но, в любом случае, применяется матричное расположение битов информации в микросхеме. Число выводов корпуса микросхемы ограничивает реализацию байтового варианта микросхемы. Для построения памяти емкостью  $64\text{К} \times 8$  требуется 8 микросхем  $64\text{К} \times 1$ , образующих одну линейку.

На рис. 14 представлена схема памяти  $1\text{К} \times 8$ , построенная на микросхемах  $1\text{К} \times 1$ .

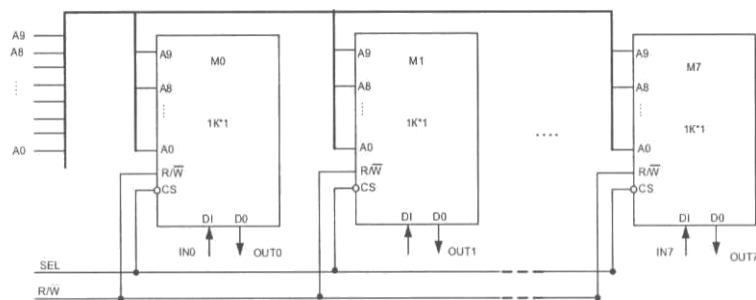


Рис. 14. Схема ЗУ 1К\*8

На схеме биты адреса, разделенные на две группы A4-A0 и A9-A5 для выбора соответственно строк и столбцов, на пересечении которых находится необходимый элемент памяти, причем входы адреса столбцов управляют формирователями и усилителями считывания.

В схеме ЗУ все адресные линии объединены в одну шину и подключены параллельно с одноименными номерами. Каждая микросхема имеет один ввод бита данных (IN) и один вывод (OUT), то есть содержит отдельные контакты. Если выбор (SEL=0) микросхем разрешен - то выполняется запись или чтение 8 битов данных, с помощью сигнала на линии R/W. При отсутствии сигнала SEL=1 микросхемы переводятся в состояние "отключено", что позволяет наращивать микросхемы, образуя матричное их соединение (несколько строк и несколько столбцов).

На рис. 15 изображена структурная схема ЗУ 1 К\* 1.

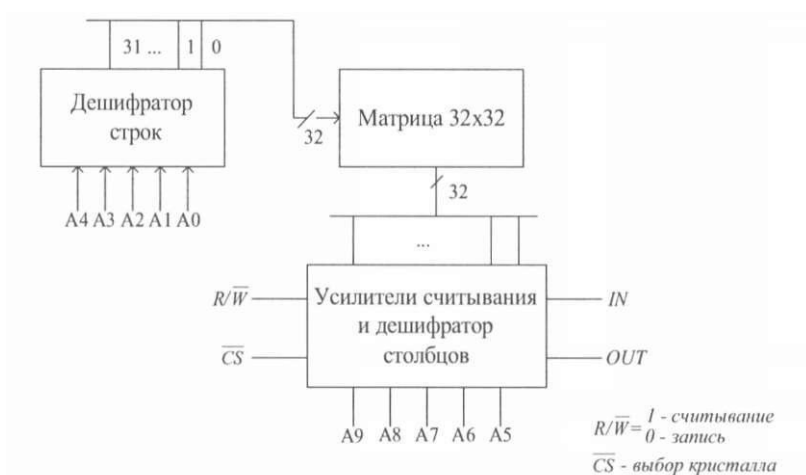


Рис. 15. Структурная схема ЗУ 1 К\* 1

На схеме биты адреса, разделенные на две группы А4-А0 и А9-А5 для выбора, соответственно, строк и столбцов, на пересечении которых находится необходимый элемент памяти, причем входы адреса столбцов управляют формирователями и усилителями считывания.

Различают в общем случае три адресных пространства: ввода-вывода (адресуемых регистров), памяти данных команд (ОЗУ) и памяти конфигурации. Причем, в периферийных микропроцессорных устройствах эти адресные пространства могут объединяться в одну общую память. В этом случае возникает задача выделения с помощью дешифраторов одной части адресов под порты ввода-вывода, другой части под ОЗУ (RAM) и еще одной части под постоянную память (ROM). В системных процессорах эти пространства разделены и для реализации этой цели используются дополнительные сигналы, указывающие к какому пространству относится тот или иной адрес команды (рис. 16).

Внешний сигнал своим низким уровнем, указывает, что обращение при выполнении ассемблерной команды MOV осуществляется к ОЗУ. Если выполняется ассемблерная команда IN или OUT, то микропроцессором формируется сигнал высокого уровня (1) и доступ выполняется к порту ввода-вывода.

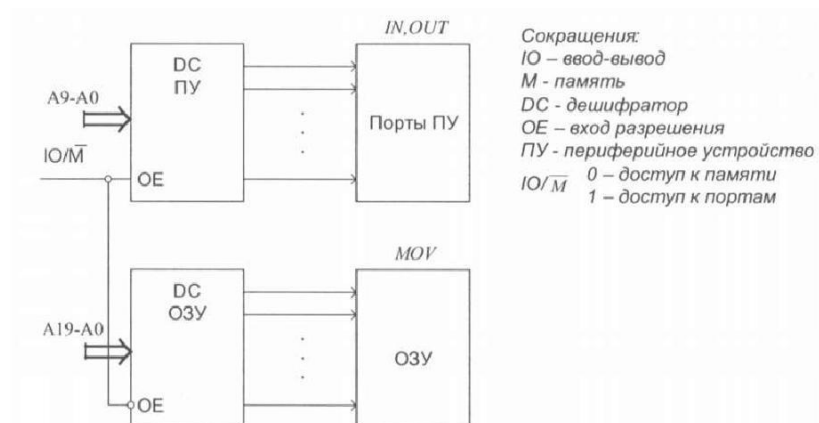


Рис. 16. Разделенные адресные пространства ПУ и ОЗУ

Рассмотрим задачу дешифрации адреса, когда в микропроцессорной системе содержится три запоминающих устройства, каждый из которых имеет емкость 16К (рис. 17).

Номер ОЗУ	Номер ячейки															16-разрядное значение		
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1		A0	
ОЗУ0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ОЗУ1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ОЗУ2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ОЗУ2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ОЗУ2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Рис. 17. Распределение адресов между ОЗУ0, ОЗУ 1 и ОЗУ2

На рис. 17 изображена схема дешифрации адресов для трех ОЗУ, построенная в соответствии с рис. 1.18.

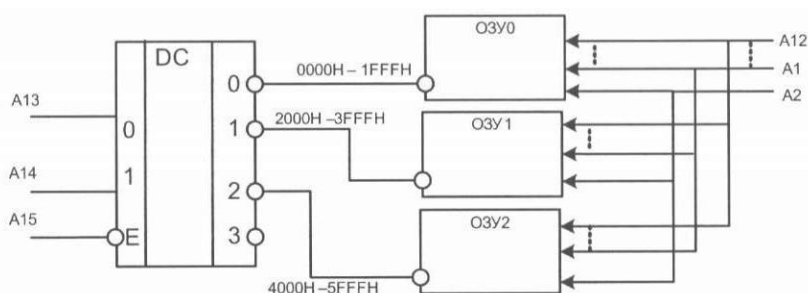


Рис. 18. Схема дешифрации адресов

Меняя значения в битах адреса A14 и A13 (00, 01 и 10), можно осуществить выбор соответствующего ОЗУ. Биты A12...A1 A0 поступают параллельно и непосредственно на адресные входы ОЗУ2, ОЗУ1, ОЗУ0 для доступа к соответствующим ячейкам памяти с помощью встроенных дешифраторов (на рис. 1.19 они не показаны).

### *Выполнение работы.*

#### Статическая память.

На рис. 19 представлена модель статической памяти, в которой реализуется описанный выше способ дешифрации адресов, когда старшая часть битов адреса используется для выбора блока памяти (БП), а младшие биты применяются для доступа к ячейкам памяти этого блока.

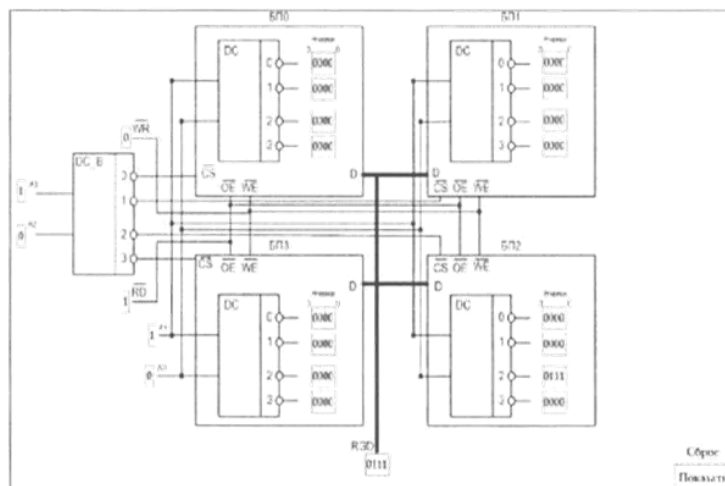


Рис. 19. Модель статической памяти с каскадной дешифрацией адресов ячеек

Модель демонстрирует принцип работы памяти с учетом диаграмм записи и чтения данных.

В режиме записи необходимо в адресных окнах ввода задать значение номера блока и номера ячейки памяти. Набрать в окне регистра данных RGD двоичный код. Затем установить активное значение сигнала записи WR равное 0. Причем значение сигнала чтения RD должно быть пассивным и равным 1. С помощью клавиши мыши нажать кнопку «Показать». На экране монитора наблюдается выбор блока памяти и ячейки с записанным в нее двоичным кодом.

Аналогичным образом осуществляется режим чтения записанной ранее двоичной информации. Устанавливается адрес соответствующего блока памяти (ячейки) и активный сигнал чтения RD=0 (WR= 1). После этого нажимается кнопка «Показать» и в окне регистра RGD наблюдается код, считанный из выбранной ячейки памяти. Предварительно окно ввода RGD должно быть очищено с помощью клавиатуры.

Динамическая память.

На рис. 20 показана упрощенная модель динамической памяти с матрицей элементов памяти 4\*4, схемой элемента памяти, содержащего конденсатор С и транзисторы, и схемами дешифрации и регенерации заряда конденсаторов.

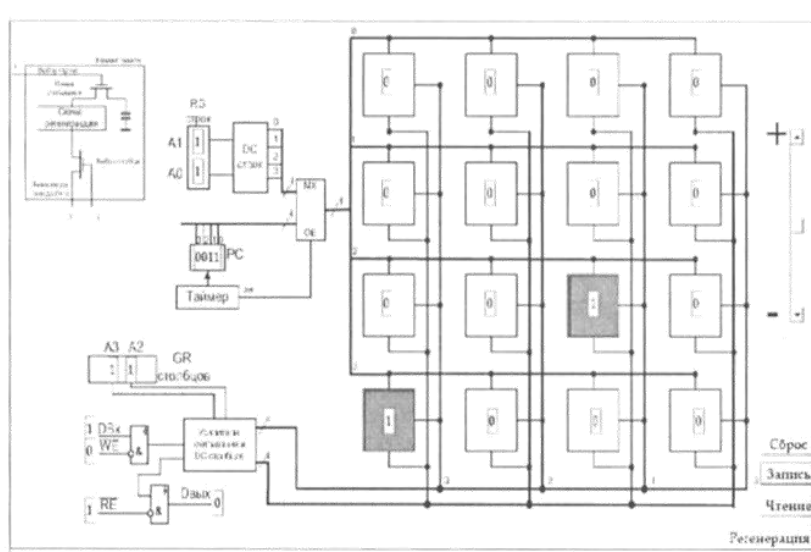


Рис. 20. Модель динамической памяти

Единичное состояние элемента памяти отмечается красным цветом (конденсатор заряжен). Модель демонстрирует режимы: запись, чтение и регенерацию памяти. В режиме записи бита устанавливаются адреса строки (A1 A0), столбца (A3 A2), значение бита на DBx (0 или 1) и активный сигнал WE=0. После этого нажимается кнопка «Запись». На экране монитора наблюдается заряд конденсатора C для единичного значения красным цветом и выбор элемента памяти и подсвечивается схема регенерации памяти в элементе памяти, демонстрируя подзарядку конденсатора C (повторная запись). Разряд конденсаторов со временем происходит по всей строке и для сохранения информации необходимо через определенные интервалы времени, определяемые таймером, с помощью счетчика СТ2 осуществляется периодическое считывание и восстановление заряда конденсатора, то есть выполняется регенерация всех строк.

Для демонстрации этого режима нажимается кнопка «Регенерация». В элементах памяти матрицы розовым цветом отмечается разряженное состояние конденсаторов, соответствующее единичным состояниям. После этого таймер запускается в работу счетчик СТ2, который через мультиплексор МХ осуществляет последовательное сканирование каждой строки. После выбора строки там, где был розовый цвет (разряженное состояние), происходит его замена на красный цвет (восстановленное состояние конденсатора). Этот процесс

повторяется до тех пор, пока не наступит сигнал «Сброс» (нажатие соответствующей кнопки). В этом режиме дешифратор (DC) столбцов не участвует, входа DBx и выхода OВых отключены ( $WE=RE=1$ ).

В режиме Чтения с помощью содержимого регистра RG строк и регистра KC столбцов выбирается элемент памяти, устанавливается активный сигнал разрешения чтения  $RE=0$  ( $WE=1$ ) и на выходе ПВых наблюдается значение выбранного элемента памяти (1 или 0). Предварительно в элементах памяти устанавливаются единичные биты (красный цвет). Предусматривается занесение первых битов путём щелчка кнопкой мыши по элементу памяти в режиме регенерации.

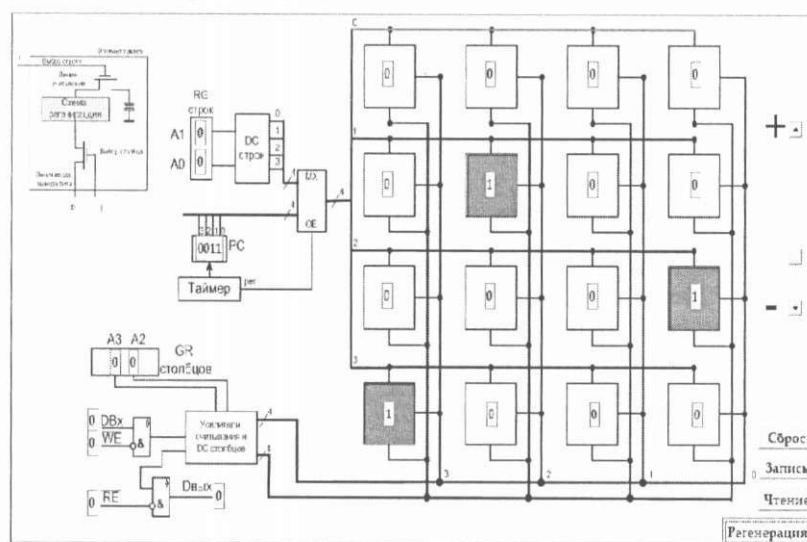


Рис 21. Пример работы с моделью динамической памяти

Имитационная модель приведена в папках «Статическая и динамическая память».

## **Практическая работа №7**

### **Диалоговые протоколы**

Практически все протоколы верхних уровней строятся на основе диалога: первая сторона отправляет запрос, вторая сторона отвечает на него. В ряде протоколов диалог может состоять из одного запроса и одного ответа (например, в протоколе HTTP), в других диалог может включать десятки запросов и ответов (например, FTP, POP3, SMTP).

#### **Протокол SMTP**

Протокол SMTP (Simple Mail Transfer Protocol, простой протокол передачи почты) отвечает за отправку почты другому адресату.

Сеанс работы протокола SMTP состоит из нескольких этапов: аутентификация, передача параметров письма (адреса, тема), отправка текста сообщения и завершение соединения.

Поскольку работа с почтой производится от имени конкретного пользователя, то первым этапом диалога идёт авторизация – клиент передаёт серверу имя пользователя и пароль. Оба протокола поддерживают безопасные методы аутентификации с применением протокола SSL, однако для данной практической работы для простоты используется нешифрованная аутентификация. Протокол SMTP кодирует пароль кодировкой base64, что обеспечивает минимальную защиту, которая, впрочем, легко вскрывается.

К письму могут прилагаться двоичные файлы, но из-за того, что протокол предусматривает передачу только ASCII-информации, файлы должны быть закодированы в текстовый формат по стандарту MIME.



Таблица 3. Основные команды протокола SMTP

Команда	Параметры	Описание
HELO	домен	Начало сессии
AUTH LOGIN		Аутентификация: сервер запросит логин и пароль, закодированные с помощью base64
MAIL FROM:	<e-mail отправителя>	Адрес отправителя письма
RCPT TO:	<e-mail получателя>	Адрес получателя письма
DATA		После этой команды сервер все данные воспринимает как тело письма. Признаком окончания тела является строка с единственной точкой.
HELP	[команда]	Вывод справочной информации по указанной команде (если она указана)
QUIT		Завершение сеанса

### Протокол POP3

Протокол POP3 (Post Office Protocol) предназначен для получения почты с почтового сервера на локальную машину.

Таблица 4. Основные команды протокола POP3:

Команда	Параметры	Описание
USER	имя пользователя	Аутентификация пользователя
PASS	пароль	Пароль открытым текстом
STAT		Получение количества сообщений в ящике и их размер
LIST	[сообщение]	Получение информации обо всех сообщениях в ящике (или о конкретном, если оно указано)
DELE	сообщение	Удаление указанного сообщения
RETR	сообщение	Загрузка указанного сообщения
QUIT		Завершение сеанса

## **Используемые утилиты**

Поскольку протоколы POP3 и SMTP основаны на текстовом диалоге между клиентом и сервером, допустимо использовать вместо специальной почтовой программы обычный клиент **telnet** или одну из совместимых с этим протоколом альтернативных программ. Для подключения к почтовому серверу нужно знать имя сервера и порт.

Имя почтового сервера может не совпадать с именем почтового адреса, например, для адреса `user@mail.ru` почтовый сервер для получения почты будет `pop.mail.ru`, а для отправки почты - `smtp.mail.ru`. Имена почтовых серверов для каждого домена обычно хранятся в записи типа MX на DNS-сервере (а при её отсутствии – используется адрес сервера из записи типа A), кроме того, их обычно можно получить в документации почтовой службы. Для протокола POP3 стандартный номер порта 110, для SMTP – 25, однако они также могут отличаться в разных службах.

Для кодирования и декодирования пароля для протокола SMTP можно использовать одну из подходящих утилит, или же один из онлайн-перекодировщиков.

### **Задание**

При выполнении практической работы потребуется подключение к почтовому серверу. Допустимо использовать любой сервер, на котором имеется учетная запись.

Используя SMTP-сервер, отправить письмо на свой адрес. Используя POP3-сервер получить количество писем в ящике, список писем с информацией о них, загрузить письмо, отправленное через SMTP-сервер.

В отчёт следует вставить полные логи сеансов связи с каждым из серверов. Для сохранения пароля к учетной записи в секрете пароль из отчета нужно удалить.

## **Практическая работа №8**

### **Программная реализация протокола высокого уровня**

#### **Задание**

Разработать приложение, выполняющее роль клиента или сервера для одного из протоколов высокого уровня:

- HTTP
- SMTP
- POP3
- FTP
- Другой (по согласованию с преподавателем)

При выполнении этого задания можно работать в паре: один человек может взять на себя написание серверной части, другой – клиентской. При одиночной работе допустимо использовать сторонние серверы или клиенты, не требующие установки или находящиеся в сети.

Недопустимо использование высокоуровневых протоколов или классов, реализующих указанную функциональность: в качестве инструмента должны использоваться только классы и процедуры для работы с сокетами, которые встроены в основную библиотеку классов выбранного языка.

В окне программ должна быть предусмотрена возможность просмотра логов: полных текстов запросов и ответов на них.

В отчете приведите исходные тексты обеих программ, а также снимки окон программ в процессе работы.

## Практическая работа №9

### Команды диагностики сети

Команды тестирования сети входят в состав сетевых операционных систем и позволяют просмотреть параметры отдельных узлов сети, проверить их работоспособность и в случае обнаружения проблемы – определить проблемный участок.

#### Используемые утилиты

Утилита `ipconfig` предназначена для отображения параметров текущей конфигурации сети TCP/IP, таких как IP-адрес, маска сети, шлюз по умолчанию, а также параметры DHCP и DNS. По умолчанию команда выводит сокращенный набор параметров, однако с помощью дополнительных ключей можно вывести полный список.

Утилита `ping` позволяет проверить наличие IP-соединения с другим узлом сети, а также время оборота пакета. Дополнительные ключи позволяют изменить количество попыток соединения, время ожидания ответа и т. д.

Утилита `tracert` позволяет определить путь до узла назначения в виде списка маршрутизаторов между исходным и конечным узлами, а также время оборота пакета. Если очередной узел не ответил в заданное время, то вместо времени оборота программа ставит звездочку. Адрес узла может быть определен только в том случае, если от него пришёл хотя бы один ответ.

Утилита `nslookup` служит для получения информации о домене. Утилита подключается к указанному DNS-серверу (если сервер не задан, то используется сервер текущего подключения) и извлекает поля требуемого типа. Если DNS-сервер ответственен за запрашиваемую зону, то ответ считается достоверным.

Утилита `netstat` предназначена для отображения активных TCP-соединений, прослушиваемых узлом портов, таблицы маршрутизации узла, а также статистики Ethernet и протоколов сетевого и транспортного уровней.

#### Задание

Использовать команды сетевой диагностики `nslookup`, `ping`, `tracert` для получения информации о двух доменах, проверки их работоспособности, отслеживания пути. Домены можно выбрать

произвольные, но не рекомендуется использовать общеизвестные домены (такие, как `google.com` или `yandex.ru`), а также домены томской сети.

Использовать команду `ipconfig` для выяснения IP-адреса компьютера, маски сети и шлюза по умолчанию, а также дополнительных параметров. Отобразить кэш DNS.

Использовать команду `netstat` для отображения статистики протоколов и открытых соединений/портов. Привести список процессов, прослушивающих порты.

Для каждой из команд следует постараться максимально использовать доступные опции.

## **Практическая работа №10**

### **Анализ пакетов локальной сети**

В обычном режиме работы сетевая карта узла обрабатывает только те пакеты, которые адресованы данному узлу, или же всем узлам сети, если пакет широковещательный. Однако существует и другой режим работы сетевой карты – так называемый *неразборчивый режим* (Promiscuous mode), в котором сетевая карта получает все проходящие по сети пакеты.

Поскольку данный режим позволяет при определенных условиях перехватывать и анализировать трафик, предназначенный для чужих узлов, использование данного режима является незаконным в ряде стран, но даже если закон не запрещает его использование явно, перехват чужих данных может привести к возникновению административной или уголовной ответственности. Тем не менее, данный режим полезен в ряде случаев администраторам сети, поскольку позволяет отследить проблемный трафик (например, широковещательный шторм) и выяснить его причину.

Анализатор трафика (sniffer, сниффер) – это программа, предназначенная для перехвата и анализа сетевого трафика.

Чтобы перевести сетевую карту в неразборчивый режим, требуются привилегии администратора локальной машины, поэтому данная практическая работа в учебной аудитории выполняется в виртуальной машине.

#### **Задание**

Выполнить захват трафика, произвести его анализ.

#### **Последовательность выполнения работы**

- 1) Запустите программу VirtualBox.
- 2) Выберите и запустите виртуальную машину Ubuntu.
- 3) Откройте терминал двойным щелчком по ярлыку на рабочем столе.
- 4) Введите команду `sudo wireshark`.
- 5) Когда терминал запросит пароль, введите `ips`.
- 6) В секции *Capture* в окне программы Wireshark выберите интерфейс `eth0`.

7) Запустите браузер FireFox и откройте какой-либо сайт, закройте браузер.

8) С помощью фильтра на панели инструментов отфильтруйте захваченный трафик, оставив только пакеты протокола TCP.

9) Скопируйте в отчёт строки анализатора трафика, относящиеся к следующим событиям (если за время захвата трафика было установлено несколько TCP-соединений, удалите из отчёта лишние строки):

- a) установка TCP-соединения;
- b) передача данных;
- c) завершение TCP-соединения.

10) Закройте Wireshark (на вопрос программы о сохранении результатов ответьте отрицательно) и другие открытые в виртуальной машине окна.

11) Завершите работу виртуальной машины: выберите в меню *Машина* пункт *Закреть...*, отметьте пункт *Сохранить состояние машины* и нажмите кнопку ОК.

12) Дождитесь сохранения состояния виртуальной машины.

13) Закройте программу VirtualBox.

## Практическая работа №11 Разбиение на подсети

Префикс сети показывает количество бит, отводимых под номер сети. Оставшиеся биты отведены под номер хоста. Например, префикс /24 говорит о том, что старшие 24 бита адреса отводятся под номер сети, а оставшиеся  $(32-24)=8$  бит – под номер хоста.

Исходя из количества бит, отводимых под номер хоста, можно определить максимальное количество компьютеров в подсети. Так, с помощью восьми бит можно закодировать 256 различных чисел, однако первое и последнее число не могут использоваться в качестве номеров хостов, поскольку им назначена особая роль. Если в номере хоста все биты равны нулю, то такой адрес называется адресом сети. Если же в номере хоста все биты равны единице, то такой адрес является широковещательным адресом для данной сети.

Пример:

IP-адрес:

192.168.1.18 /24

В двоичном виде:

11000000 10101000 00000001 00010010

Номер сети:

11000000 10101000 00000001 00000000

Широковещат. адрес:

11000000 10101000 00000001 11111111

Кроме номера сети и широковещательного адреса подсеть обычно содержит шлюз, роль которого, как правило, выполняет один из интерфейсов маршрутизатора. Часто шлюз получает первый либо последний адрес сети, который можно назначить хосту.

Подсеть разбивается на более мелкие подсети увеличением префикса подсети. Поскольку префикс задаёт количество бит в номере сети, такая операция всегда делит общее количество адресов подсети, на число, кратное степени двойки.



Пример:

Исходная подсеть:

192.168.1.0 /24

В двоичном виде:

11000000 10101000 00000001 00000000

Новые подсети:

192.168.1.0 /25

192.168.1.128 /25

В двоичном виде:

11000000 10101000 00000001 00000000

11000000 10101000 00000001 10000000

Таким образом, процесс разбиения исходной сети на нужное количество подсетей сводится к дихотомическому делению исходной подсети до тех пор, пока для каждого набора компьютеров не будет найдена сеть минимально необходимого размера.

Каждая пара маршрутизаторов также соединяется между собой отдельной подсетью. Поскольку такая подсеть содержит только два хоста, для неё достаточно использовать префикс подсети /30.

### Методика выполнения

Исходную сеть и её подсети можно представить в виде графа, вершиной которого будет исходная сеть, а каждый дочерний узел получается в результате разбиения сети на две подсети:

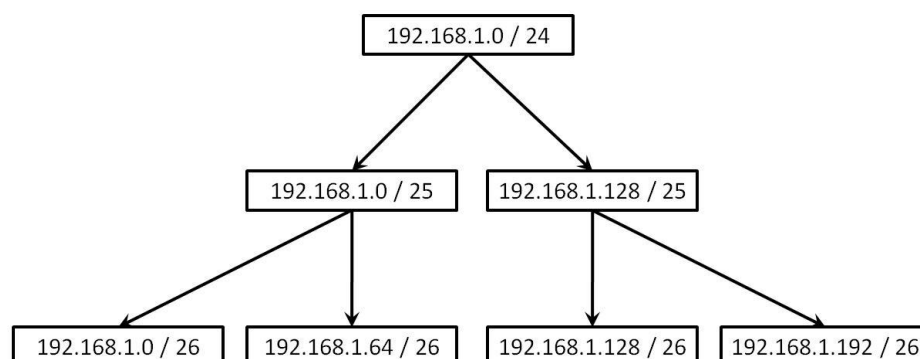


Рисунок 22 - Граф подсетей

Деление продолжается до тех пор, пока для каждой физической подсети не будет найдена подходящая по размерам подсеть.

Другой наглядный метод разбиения на подсети заключается в использовании программ для работы с электронными таблицами, например, Microsoft Excel. Каждая ячейка представляет некоторый диапазон адресов, например, подсеть с префиксом /30. Объединенные ячейки представляют собой более крупные сети, которые затем можно делить пополам в следующих столбцах:

	1	2	3	4	5	6
1	192.168.1.0 / 25	192.168.1.0 / 26	192.168.1.0 / 27	192.168.1.0 / 28	192.168.1.0 / 29	192.168.1.0 / 30
2						192.168.1.4 / 30
3					192.168.1.8 / 29	192.168.1.8 / 30
4						192.168.1.12 / 30
5				192.168.1.16 / 28	192.168.1.16 / 29	192.168.1.16 / 30
6						192.168.1.20 / 30
7					192.168.1.24 / 29	192.168.1.24 / 30
8						192.168.1.28 / 30
9			192.168.1.32 / 27	192.168.1.32 / 28	192.168.1.32 / 29	192.168.1.32 / 30
10						192.168.1.36 / 30
11					192.168.1.40 / 29	192.168.1.40 / 30
12						192.168.1.44 / 30
13				192.168.1.48 / 28	192.168.1.48 / 29	192.168.1.48 / 30
14						192.168.1.52 / 30
15					192.168.1.56 / 29	192.168.1.56 / 30
16						192.168.1.60 / 30
17		192.168.1.64 / 26	192.168.1.64 / 27	192.168.1.64 / 28	192.168.1.64 / 29	192.168.1.64 / 30
18						192.168.1.68 / 30
19					192.168.1.72 / 29	192.168.1.72 / 30
20						192.168.1.76 / 30
21				192.168.1.80 / 28	192.168.1.80 / 29	192.168.1.80 / 30
22						192.168.1.84 / 30
23					192.168.1.88 / 29	192.168.1.88 / 30
24						192.168.1.92 / 30
25			192.168.1.96 / 27	192.168.1.96 / 28	192.168.1.96 / 29	192.168.1.96 / 30
26						192.168.1.100 / 30
27					192.168.1.104 / 29	192.168.1.104 / 30
28						192.168.1.108 / 30
29				192.168.1.112 / 28	192.168.1.112 / 29	192.168.1.112 / 30
30						192.168.1.116 / 30
31					192.168.1.120 / 29	192.168.1.120 / 30
32						192.168.1.124 / 30

Рисунок 23 - Разбиение на подсети в Microsoft Excel

Разумеется, оба эти способа позволяют узнать лишь начальный адрес подсети, остальные данные необходимо рассчитать на основе этих данных.

## Задание

Используя схему сети, приведенную на **Ошибка! Источник ссылки не найден.**, а также информацию о количестве компьютеров в отделах предприятия (таблица ), разбейте сеть на соответствующее количество подсетей. Разбиение должно быть оптимальным, то есть не следует использовать для отдела подсеть, если достаточно будет половины подсети. В отчете приведите:

1. Схему сети с подписанными подсетями.
2. Параметры каждой подсети:
  - 1) адрес сети (в двоичном и десятичном виде);
  - 2) префикс;
  - 3) маска (в двоичном и десятичном виде);
  - 4) широковещательный адрес
  - 5) адрес шлюза;
  - 6) максимальное количество хостов;
  - 7) количество неиспользуемых адресов хостов.

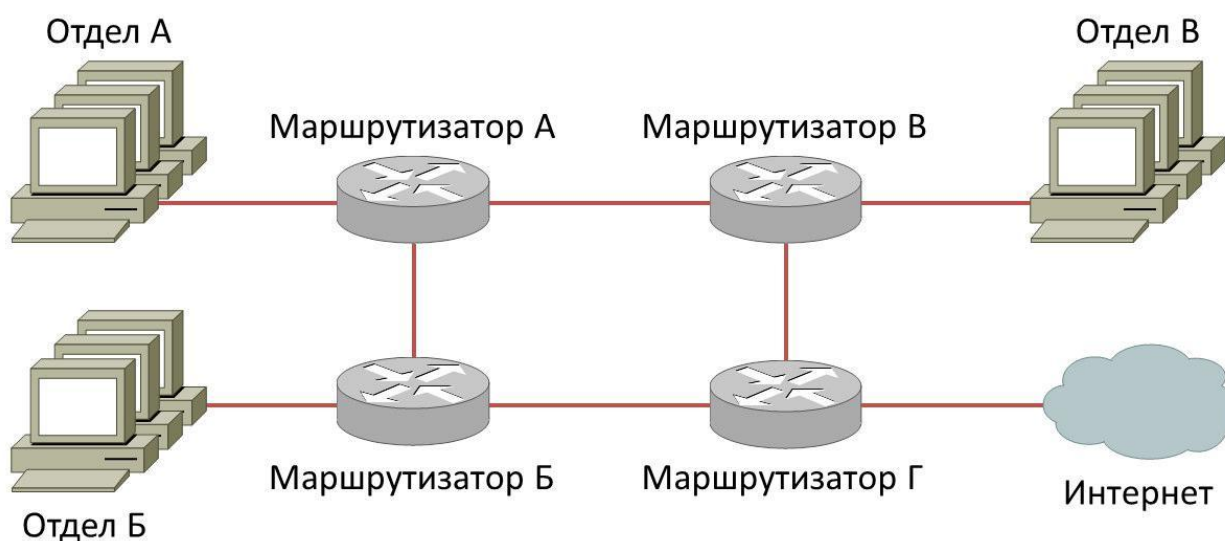


Рисунок 24 - Схема сети предприятия

Таблица 5. Варианты заданий

№	Исходная сеть	Количество компьютеров в отделах		
		А	Б	В
1	34.178.0.0 /16	3750	6793	1702
2	118.7.50.0 /24	7	9	27
3	39.221.98.0 /24	8	5	18
4	88.27.252.0 /23	30	9	46
5	81.104.216.0 /21	48	120	249
6	7.50.128.0 /19	267	176	678
7	89.151.32.0 /19	311	246	806
8	126.61.74.0 /23	8	61	17
9	36.121.96.0 /19	311	696	226
10	28.54.64.0 /19	957	153	274
11	67.253.0.0 /16	3656	1165	5086
12	77.75.0.0 /18	338	830	1403
13	5.63.168.0 /21	119	61	226
14	85.123.72.0 /21	189	51	72
15	72.241.3.0 /25	12	7	3
16	87.228.68.0 /22	26	45	71
17	46.41.64.0 /18	384	1535	675
18	57.214.86.0 /23	63	9	21
19	74.30.128.0 /19	346	179	732
20	88.61.128.0 /20	366	77	130
21	10.58.180.0 /22	30	92	43
22	112.56.76.0 /22	23	114	60
23	2.78.160.0 /19	214	443	525
24	30.182.64.0 /18	624	1700	358
25	75.39.128.0 /19	625	219	372

## Практическая работа №12

### Статическая маршрутизация

#### Подключение через терминал

Для административного подключения к маршрутизатору доступны несколько способов: через интернет, через телефонный канал, через терминал. Если маршрутизатор находится в физически доступном месте, то самым простым способом для подключения и конфигурирования будет использование терминала. В качестве эмулятора терминала можно использовать любую программу, которая обеспечивает подключение к устройству через СОМ-порт, например, `HyperTerminal` или `Putty`. Для физического подключения компьютера к маршрутизатору Cisco используется специальный консольный кабель Cisco голубого цвета.

После запуска программы в настройках терминальной сессии следует установить ряд параметров, которые требуются для нормальной работы:

- Подключение: COM1
- Скорость бит: 9600
- Биты данных: 8
- Четность: нет
- Стоповые биты: 1
- Управление потоком: нет

Большинство эмуляторов терминала позволяют сохранить установки этих параметров и использовать их при последующих подключениях.

#### Режимы работы Cisco IOS

Для обеспечения безопасности Cisco IOS предоставляет четыре различных режима работы. Каждый из режимов позволяет выполнять определенные множества команд, и визуально отличается приглашением командной строки.

#### *Пользовательский режим*

Данный режим предоставляет ограниченные возможности, в нём разрешены лишь некоторые базовые операции для мониторинга. Он не позволяет выполнять никакие команды, которые могут изменить конфигурацию устройства. Приглашение командной строки в этом режиме выглядит следующим образом:

Router>

Вместо слова **Router** слева от знака режима может стоять другое имя, заданное пользователем.

### ***Привилегированный режим***

Выполнение конфигурационных или управляющих команд требует вхождения в привилегированный режим. Командная строка в этом режиме выглядит так:

Router#

Для перехода в привилегированный режим используется команда **enable**, для выхода – **disable**.

В рабочих сетях вход привилегированный режим может требовать авторизации пользователя. Изначально маршрутизаторы поставляются либо без пароля, либо с логином и паролем **cisco / cisco**.

### ***Режим глобального конфигурирования***

Из режима глобальной конфигурации можно делать изменения, который касаются устройства в целом. Также данный режим позволяет входить в режим конфигурирования определенного интерфейса. Приглашение командной строки в этом режиме выглядит так:

Router(config)#

Для входа в режим глобального конфигурирования используется команда **configure** с последующим указанием способа передачи конфигурационной информации. Например, если конфигурирование осуществляется с терминала, то команда будет выглядеть следующим образом:

Router#**configure terminal**

Вход в режим глобального конфигурирования возможен только из привилегированного режима.

### ***Режим конфигурирования интерфейса***

Данный режим представляет собой одно из подмножеств режима глобального конфигурирования и позволяет настраивать один из доступных сетевых интерфейсов (**Fa0/0**, **S0/0/0** и т. д.) Помимо режима конфигурирования интерфейса существуют также режимы конфигурирования линий (физических или виртуальных), а также режимы конфигурирования протоколов маршрутизатора. Все изменения, вносимые в конфигурацию маршрутизатора в

данном режиме относятся только к выбранному интерфейсу. Приглашение командной строки в этом режиме выглядит следующим образом:

```
Router(config-if)#
```

Для входа в данный режим используется команда `interface`. Например, для конфигурирования интерфейса `FastEthernet 0/0` следует выполнить команду

```
Router(config)#interface fastethernet 0/0
```

### Команды Cisco IOS

Cisco IOS предоставляет администратору широкий набор команд для управления работой маршрутизатора и его диагностики. При выборе команды следует помнить, что каждая из них может быть предназначена для работы только в определенном режиме, при попытке выполнить данную команду в другом режиме Cisco IOS её не опознает. Например, команда `hostname`, которая позволяет изменить внутреннее имя маршрутизатора, может быть выполнена только в режиме глобального конфигурирования:

```
Router>enable
```

```
Router#config t
```

```
Router(config)#hostname RtA
```

```
RtA(config)#
```

Действие ряда команд может быть отменено выполнением команды со специальной приставкой `no`, например, для отмены переименования маршрутизатора следует выполнить такую команду:

```
RtA(config)#no hostname
```

```
Router(config)#
```

Ещё одна важная команда, которая имеет множество дополнительных параметров и используется в разных режимах – это команда `show`. Например, чтобы посмотреть текущую конфигурацию маршрутизатора, следует выполнить команду

```
Router#show running-config
```

Cisco IOS отобразит ту конфигурацию, которая является актуальной в данный момент. Для просмотра конфигурации маршрутизатора, которая будет применена в случае перезапуска

или сбоя питания, служит та же команда `show`, только с другим параметром:

```
Router#show startup-config
```

Выводимая этой и другими командами информация может не помещаться в окне терминала. В этом случае Cisco IOS приостановит вывод и напечатает строчку

```
--More--
```

В этом случае следует нажать либо клавишу Пробел, которая выведет следующие строки на терминал, либо `Enter`, которая выведет только одну следующую строку. Нажатие комбинации `Ctrl+C` прервёт вывод текста и вернёт приглашение Cisco IOS.

Большинство команд Cisco IOS могут быть сокращены до нескольких первых букв: так, вместо `configure terminal` можно писать `config t`, а вместо `fastethernet` – просто `fa`. Сокращения ускоряют работу, однако усложняют понимание, поэтому не стоит чрезмерно ими увлекаться.

### **Конфигурация интерфейса**

Как уже отмечалось выше, режим конфигурирования интерфейса позволяет управлять каждым интерфейсом независимо от остальных. Для входа в данный режим следует использовать команду `interface` в одном из следующих вариантов:

```
Router#interface тип порт
```

```
Router#interface тип слот/порт
```

```
Router#interface тип слот/подслот/порт
```

Обычно для физических интерфейсов обозначения типа, слота, подслота и порта написаны рядом с самим интерфейсом. Для выключения интерфейса служит команда `shutdown`, для повторного включения – команда `no shutdown`:

```
Router(config-if)#shutdown
```

```
Router(config-if)#no shutdown
```

Интерфейсу может быть назначен IP-адрес и маска сети:

```
Router(config-if)#ip address 192.168.1.1 255.255.255.0
```

Если интерфейсы двух маршрутизаторов соединены между собой, то IP-адреса каждого из интерфейсов должны находиться в одной подсети!

Для выхода из режима конфигурации интерфейса используется команда `exit`:



```
Router(config-if)#exit
Router(config)#
```

### Команды мониторинга

Для проверки связи между устройствами сети можно использовать команды `ping` и `tracert`:

```
Router#ping 192.168.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echoes to 192.168.1.2, timeout is 2 seconds:

```
!!!!
```

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Каждый ICMP-пакет, на который был получен ответ, обозначается восклицательным знаком, каждый потерянный пакет – точкой.

```
Router#tracert 192.168.1.2
```

Type escape sequence to abort.

```
Tracing the route to 192.168.1.2
```

```
 0 172.16.0.253 8 msec 4 msec 8 msec
```

```
 1 10.0.0.254 16 msec 16 msec 8 msec
```

```
 2 192.168.1.1 16 msec * 20 msec
```

Для проверки состояния интерфейса можно использовать команду `show` с дополнительными параметрами:

```
Router#show ip interface brief
```

```
Interface      IP-Address OK? Meth. Status Protocol
```

```
FastEthernet0/0 192.168.1.1 YES NVRAM up up
```

```
Serial0/0/1 unassigned YES unset adm. down down
```

Глядя на вывод данной команды (часть текста опущена) можно определить, что интерфейсу `FastEthernet0/0` назначен IP-адрес `192.168.1.1`, последние две колонки определяют статус интерфейса – у включенного работоспособного интерфейса оба параметра должны иметь значение `up`. Значение `administratively down` означает, что интерфейс был отключен администратором.

### Статическая маршрутизация

Статическая маршрутизация предполагает фиксированную структуру сети: каждый маршрутизатор в сети точно знает куда

нужно отправлять пакет, чтобы он был доставлен по назначению. С одной стороны, это упрощает работу администратора в небольших сетях – достаточно один раз прописать все маршруты; такая конфигурация не будет создавать дополнительный служебный трафик внутри сети. С другой стороны, если сеть достаточно большая, то ручная настройка всех маршрутизаторов является утомительным делом, не исключены и ошибки. Более того, при изменении топологии сети нужно будет вручную переконфигурировать каждое из сетевых устройств.

Рассмотрим статическую маршрутизацию на примере. Если два маршрутизатора соединены между собой, то каждый из них знает о том, какая сеть подключена непосредственно к нему.

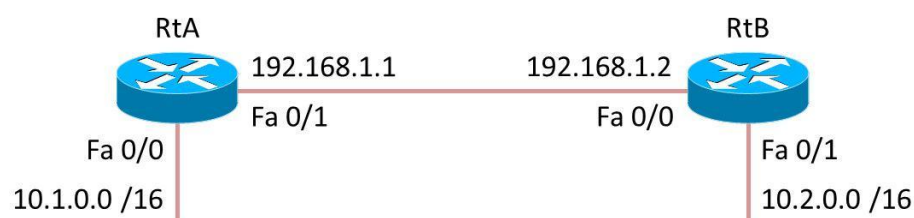


Рисунок 25 - Сеть из двух маршрутизаторов

Так, маршрутизатор RtA знает, что для отправки пакета в сеть 10.1.0.0/16 нужно использовать интерфейс Fa 0/0, а сеть 192.168.1.0/24 подключена непосредственно к интерфейсу Fa 0/1, однако он ничего не знает о сети 10.2.0.0/16. Для того, чтобы пакеты могли быть доставлены в правильную сеть, нужно сообщить маршрутизатору RtA, что они должны быть отправлены на маршрутизатор RtB, который, в свою очередь, знает о сети RtB. Делается это с помощью команды `ip route`, в качестве параметра которой указывается либо соответствующий *выходной* интерфейс текущего маршрутизатора, либо адрес *входного* интерфейса маршрутизатора назначения. Таким образом, обе следующие команды эквивалентны:

```

RtA(config)#ip route 10.2.0.0 255.255.0.0 192.168.1.2
RtA(config)#ip route 10.2.0.0 255.255.0.0 Fa0/1
  
```

Однако если теперь попробовать из сети 10.1.0.0/16 выполнить команду `ping` с адресом назначения в сети 10.2.0.0/16, то результаты по-прежнему будут неудовлетворительными. Это происходит потому, что пакеты в

сеть назначения доставляются, но ответы обратно не доходят, ведь маршрутизатор RтВ ничего не знает о сети 10.1.0.0/16, поэтому для маршрутизатора RтВ также необходимо прописать статические маршруты.

Для хранения информации об известных маршрутах маршрутизатор использует таблицу маршрутизации. Просмотреть её можно с помощью следующей команды:

**RtA#show ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP

B – BGP, D - EIGRP, EX - EIGRP external, O – OSPF

IA - OSPF inter area, N1 - OSPF NSSA external type 1

N2 - OSPF NSSA external type 2, E1 - OSPF external type

E2 - OSPF external type 2, E – EGP, i – IS-IS

L1 - IS-IS level-1, L2 - IS-IS level-2, ia

- IS-IS inter area, \* - candidate default U

- per-user static route, o - ODR P -

periodic downloaded static route

Gateway of last resort is not set

S 192.168.1.0/24 is directly connected, FastEthernet0/1

C 192.168.2.0/24 is directly connected, FastEthernet0/1

C 192.168.3.0/24 is directly connected, FastEthernet0/0

В первой части вывода этой команды приведена легенда – обозначения кодов каждой записи из таблицы маршрутизатора. Например, код С означает, что данная сеть непосредственно подключена к одному из интерфейсов, код S – что данный маршрут был добавлен вручную (поэтому он называется статическим), код R – что данный маршрут был получен в результате работы протокола RIP и т.д. Ниже идут непосредственно сами маршруты: для каждого из них указана сеть назначения, количество бит сети в адресе, а также интерфейс, в который будет отправлен пакет для доставки в нужную сеть.

### **Задание**

Соединить и настроить маршрутизаторы для работы в сети со следующей топологией:

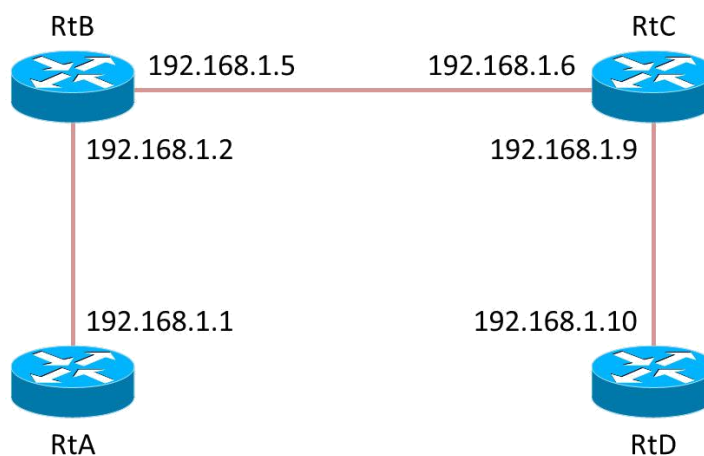


Рисунок 26- Сеть из четырёх маршрутизаторов

Пошаговая инструкция:

1. Соедините соответствующие порты маршрутизаторов перекрестными кабелями.
2. Запустите терминальную программу, например, HyperTerminal и откройте терминальную сессию с нужными параметрами.
3. Подключите консольный кабель к первому маршрутизатору.
4. Пользуясь терминалом:
  - 1) войдите в режим глобальной конфигурации;
  - 2) измените имя маршрутизатора на **RtA**;
  - 3) настройте интерфейс, к которому подключен соседний маршрутизатор:
    - a) войдите в режим конфигурирования интерфейса;
    - b) задайте IP-адрес для данного интерфейса;
    - c) активируйте интерфейс;
    - d) выйдите из режима конфигурирования интерфейса;
  - 4) если у маршрутизатора используются другие интерфейсы, то повторите шаг с для каждого из них;
  - 5) пропишите статические пути для каждой сети, которая не является соседней для данного маршрутизатора;
  - 6) выйдите из режима глобальной конфигурации;
5. Повторите пункты 3-4 для каждого маршрутизатора.
6. Выполните проверку связи между маршрутизаторами

RtA и RtD в обоих направлениях с помощью команд `ping` и `tracert`.

7. В отчете отразите следующую информацию по каждому маршрутизатору:

- 1) команды, необходимые для конфигурации, с пояснениями сути каждой команды;
- 2) таблицу маршрутизации;
- 3) результаты выполнения команд `ping` и `tracert`.

## **Практическая работа №13**

### **Динамическая маршрутизация**

Динамическая маршрутизация – это другой способ настройки сети. При этом способе не требуется прописывать вручную все пути к другим сетям, достаточно указать адреса интерфейсов и включить выбранный протокол динамической маршрутизации – после этого маршрутизаторы сами будут обмениваться информацией о маршрутах и поддерживать свои таблицы маршрутизации в актуальном состоянии.

Протокол маршрутизации RIP предназначен для сравнительно небольших и относительно однородных сетей. Маршрут характеризуется вектором расстояния до места назначения. Предполагается, что каждый маршрутизатор является отправной точкой нескольких маршрутов до сетей, с которыми он связан.

Описания этих маршрутов хранятся в специальной таблице, называемой таблицей маршрутизации, которая содержит по одной записи на каждый маршрут. Запись должна включать в себя:

- 1) IP-адрес места назначения;
- 2) метрика маршрута (от 1 до 15 шагов до места назначения);
- 3) IP-адрес ближайшего маршрутизатора по пути к месту назначения;
- 4) таймеры маршрута.

Раз в 30 секунд каждый маршрутизатор посылает широковещательно копию своей маршрутной таблицы всем соседям-маршрутизаторам, с которыми связан непосредственно. Маршрутизатор-получатель просматривает таблицу. Если в таблице присутствует новый путь или сообщение о более коротком маршруте, или произошли изменения длин пути, эти изменения фиксируются получателем в своей маршрутной таблице. Если между отправителем и приемником расположено три маршрутизатора, считается, что между ними 4 шага.

Протокол RIP не работает с адресами подсетей – он основан на классовой адресации. Протокол RIPv2 является новой версией протокола RIP, которая позволяет работать с масками подсетей.

## Команды

Перед включением протокола динамической маршрутизации нужно обычным образом сконфигурировать интерфейсы, назначив им адреса. Для включения протокола RIPv2 в режиме конфигурации нужно выполнить команду `RtA(config)#router rip`

После выполнения данной команды маршрутизатор перейдёт в режим конфигурации маршрутизатора, где можно задать дополнительные параметры.

`RtA(config-router)#version 2`

Для каждой подключенной сети нужно выполнить команду `network`, которая заставит протокол RIP работать с этой сетью.

`RtA(config-router)#network 192.168.1.0`

Маршрутизатор позволяет включить режим отладки, чтобы отслеживать изменения состояний протокола RIPv2:

`RtA#debug ip route`

## Задание

Соединить и настроить маршрутизаторы для работы в сети со следующей топологией:

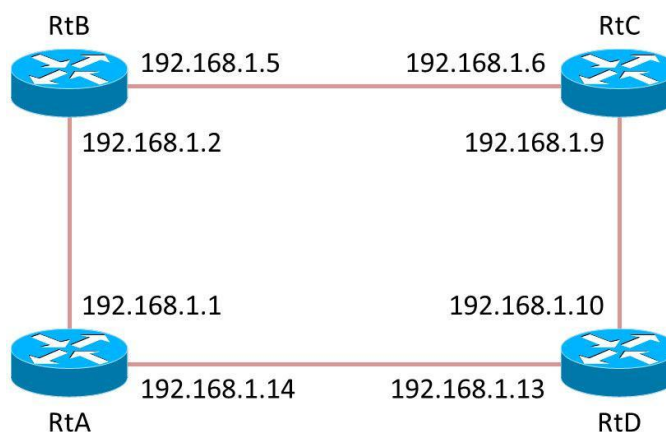


Рисунок 27 - Сеть из четырёх маршрутизаторов

После того, как сеть окажется в рабочем состоянии, отразите в отчёте все таблицы маршрутизации и результаты трассировок путей. После этого включите режим отладки протокола RIPv2, отключите один из кабелей, дождитесь перестройки таблиц маршрутизации и отразите в отчете новые таблицы и результаты трассировки.

## **Библиография**

1. Цилькер Б.Я. Организация ЭВМ и систем: Учебник для вузов / Б.Я. Цилькер, С.А. Орлов. - СПб.: Питер, 2011. - 688 с.
2. Авдеев В.А. Организация ЭВМ и периферия / В.А. Авдеев. - М.: ДМК Пресс, 2014. - 708 с.
3. Пятибратов А.П., Гудыно Л.П., Кириченко А.А. Вычислительные системы, сети и телекоммуникации: учебное пособие. – М.: КНОРУС, 2013. – 317 с.
4. Олифер В. Г., Олифер Н. А. Основы компьютерных сетей: учебное пособие. – СПб.: Питер, 2014. – 352 с.



## ПРИЛОЖЕНИЕ А

### Содержание отчета

Перед выполнением практической работы студент обязан ознакомиться с теоретическим материалом по данной теме.

**Отчет по практической работе должен содержать:**

- 1) тему работы;
- 2) условие задания;
- 3) ход выполнения работы;
- 4) результаты выполнения работы;
- 5) выводы.

Практическая работа считается выполненной, если все задания выполнены, получены соответствующие результаты, составлен полный отчет по указанной форме. После выполнения студент допускается к защите практической работы.

На защиту выносятся:

- 1) теоретические сведения по теме данной практической работы;
- 2) результаты выполнения работы.

Практическая работа считается выполненной, если она выполнена и защищена.